

Escuela Politécnica Superior



Ingeniería Técnica Industrial Mecánica

Departamento de Ingeniería Mecánica

PROYECTO FIN DE CARRERA

**ANÁLISIS DE UN ALGORITMO DE CONTROL PARA UN
ROBOT ANDADOR MEDIANTE SIMULACIÓN CON
ADAMS**

Autor: Víctor Martín Torrejón

Tutores: Jesús Meneses Alonso

Mauricio Alba Lucero

Octubre 2011



UNIVERSIDAD CARLOS III DE MADRID



Índice General

1. Objetivos	1
2. Estado del arte	3
2.1. Los primeros autómatas	4
2.2. Robots bípedos.....	5
2.2.1. Robots bípedos pasivos.....	5
2.2.2. Robots bípedos activos.....	8
3. Diseño del robot.....	13
3.1. Descripción del movimiento	14
3.2. Diseño del robot.....	15
4. Estudio teórico del robot	17
4.1 Características geométricas.....	18
4.2 Estudio cinemático	19
4.3 Estudio dinámico	20
4.4 Ecuaciones de control	23
4.5.1. Pata en el aire	23
4.5.2. Pata apoyada	24
4.5 Cálculo de constantes y variables	24
4.5.1 Cálculo de las masas de los componentes	24
4.5.2 Cálculo de los centros de masas	25
4.5.3 Cálculo de las inercias de los componentes.....	26
4.6 Ecuaciones de control finales	28
4.6.1 Pata en el aire	28
4.6.2 Pata apoyada	29
5. Modelado y simulación con Adams View.....	30
5.1. Introducción al manejo de Adams View	31
5.1.1 Botones básicos.....	32
5.1.2 Diseño de piezas.....	33
5.1.3 Diseño de juntas y generadores de movimiento	35
5.1.4 Aplicación de fuerzas y contactos	36
5.1.5 Simulación	37
5.1.6 Toma de datos	38
5.1.7 Postprocesador	38
5.2. Modelado del robot	41
5.2.1 Cambio en el diseño	41
5.2.2 Modelado	42
5.2.2.1 Creación de puntos	42
5.2.2.2 Creación de piezas.....	43
5.2.2.3 Uniones entre piezas	44
5.2.2.4 Plataformas y contactos.....	45
5.2.3. Introducción de pares	47
5.2.3.1 Pata en el aire	51
5.2.3.2 Pata apoyada	55
6. Resultados de las simulaciones.....	59
6.1 Simulaciones variando el periodo de control.....	60



6.1.1 Periodo de 0'5 segundos.....	60
6.1.2 Periodo de 0'75 segundos.....	61
6.1.3 Periodo de 1 segundos.....	63
6.1.4 Periodo de 1'25 segundos.....	64
6.1.5 Periodo de 1'5 segundos.....	65
6.1.6 Periodo de 2 segundos.....	66
6.1.7 Resumen de los datos obtenidos en las simulaciones	67
6.2 Simulaciones variando la amplitud de control	70
6.2.1 Amplitud de 15°	70
6.2.2 Amplitud de 30°	71
6.2.3 Amplitud de 45°	72
6.2.4 Amplitud de 60°	73
6.2.5 Amplitud de 75°	74
6.2.6 Amplitud de 90°	75
6.2.7 Resumen de los datos obtenidos en las simulaciones	76
6.3 Simulaciones variando el coeficiente de rozamiento	80
6.4 Simulación con doble paso.....	82
6.4.1 Introducción de nuevos pares	82
6.4.2 Doble paso con cambio en función del tiempo.....	84
7. Conclusiones	88
8. Futuros trabajos	91
9. Bibliografía e infografía.....	93
10. Anexo	95



Índice de figuras

Figura 2.1: Dynamite “Passive Dynamic Walker” con rodillas de McGeer y bípedo tridimensional de la Universidad de Cornell	6
Figura 2.2: Robot Ranger de la Universidad de Cornell	6
Figura 2.3: Robot Toddler del MIT	7
Figura 2.4: Robot Denise de la Universidad de Delft	7
Figura 2.5: Wabot-1 de la Universidad de Waseda	9
Figura 2.6: Wabot-2 de la Universidad de Waseda	9
Figura 2.7: Evolución de los robots de Honda y Asimo	10
Figura 2.8: QRIO de Sony	10
Figura 2.9: HPR-2P	11
Figura 2.10: Wabian 2 de la Universidad de Waseda	11
Figura 3.1: Posiciones de cruce entre patas	15
Figura 3.2: Medidas de pata del modelo	16
Figura 3.3: Medidas del disco del modelo	16
Figura 4.1: Descripción de ángulos del modelo	18
Figura 4.2: Reacciones y momentos en las piezas en el estudio dinámico	20
Figura 4.3: Descripción de distancias del modelo	25
Figura 5.1: Menú inicio Adams View	31
Figura 5.2: Menú creación de piezas	33
Figura 5.3: Tabla de puntos	33
Figura 5.4: Menú independiente de creación de piezas	34
Figura 5.5: Menú de creación de juntas y generadores de movimiento	35
Figura 5.6: Menú independiente de creación de juntas y generadores de movimiento	35
Figura 5.7: Menú de creación de fuerzas y contactos	36
Figura 5.8: Menú independiente de fuerzas y contactos	36
Figura 5.9: Creación de contacto	36
Figura 5.10: Creación de par	37
Figura 5.11: Menú de simulación	37
Figura 5.12: Paso de medida del postprocesador	38
Figura 5.13: Postprocesador	39
Figura 5.14: Exportación de datos y archivos	39
Figura 5.15: Carga y visualización de animaciones en el postprocesador	40
Figura 5.16: Grabación de animaciones	40
Figura 5.17: Nuevo diseño del robot	41
Figura 5.18: Tabla de puntos	42
Figura 5.19: Creación de patas	43
Figura 5.20: Creación del disco	43
Figura 5.21: Vista tridimensional del conjunto	43
Figura 5.22: Uniones entre piezas	44
Figura 5.23: Puntos guía para las plataformas	45



Figura 5.24: Creación y vista tridimensional de las plataformas.....	45
Figura 5.25: Creación de contactos.....	46
Figura 5.26: Contactos entre piezas y plataformas	46
Figura 5.27: Menú unidades.....	47
Figura 5.28: Menú creación de pares.....	48
Figura 5.29: Características de los pares Tf y Th.....	48
Figura 5.30: Ruta para crear una función	49
Figura 5.31: Menú creación de función	49
Figura 5.32: Menú creación de par	50
Figura 5.33: Menú de mediciones básicas	50
Figura 5.34: Menú avanzado de mediciones	51
Figura 5.35: Creación de la función I_{leg}	52
Figura 5.36: Creación de la función θ	53
Figura 5.37: Criterio de signos seguido	54
Figura 5.38: Creación de la función Tf.....	54
Figura 5.39: Creación de la función γ	56
Figura 5.40: Medida de la velocidad angular $\dot{\gamma}$	56
Figura 5.41: Medida del par Tf.....	57
Figura 5.38: Creación de la función Th.....	58
Figura 6.1: Simulación para $t_s = 0'5s$ $\theta_s = 1'0472rad$	60
Figura 6.2: Resultados de ángulos para la simulación $t_s = 0'5s$ $\theta_s = 1'0472rad$.61	
Figura 6.3: Simulación para $t_s = 0'75s$ $\theta_s = 1'0472rad$	62
Figura 6.4: Resultados de ángulos para la simulación $t_s = 0'75s$ $\theta_s =$ $1'0472rad$	62
Figura 6.5: Simulación para $t_s = 1s$ $\theta_s = 1'0472rad$	63
Figura 6.6: Resultados de ángulos para la simulación $t_s = 1s$ $\theta_s = 1'0472rad$...63	
Figura 6.7: Simulación para $t_s = 1'25s$ $\theta_s = 1'0472rad$	64
Figura 6.8: Resultados de ángulos para la simulación $t_s = 1'25s$ $\theta_s =$ $1'0472rad$	64
Figura 6.9: Simulación para $t_s = 1'5s$ $\theta_s = 1'0472rad$	65
Figura 6.10: Resultados de ángulos para la simulación $t_s = 1'5s$ $\theta_s =$ $1'0472rad$	65
Figura 6.11: Simulación para $t_s = 2s$ $\theta_s = 1'0472rad$	66
Figura 6.12: Resultados de ángulos para la simulación $t_s = 2s$ $\theta_s = 1'0472rad$..66	
Figura 6.13: Representación gráfica de los errores de periodos	68
Figura 6.14: Representación gráfica de los errores de θ	68
Figura 6.15: Representación gráfica de los errores de γ	69
Figura 6.16: Representación gráfica de los errores de distancias.....	69
Figura 6.17: Simulación para $t_s = 0'75s$ $\theta_s = 0'2618rad$	70
Figura 6.18: Resultados de ángulos para la simulación $t_s = 0'75s$ $\theta_s =$ $0'2618rad$	71
Figura 6.19: Simulación para $t_s = 0'75s$ $\theta_s = 0'5236rad$	71
Figura 6.20: Resultados de ángulos para la simulación $t_s = 0'75s$ $\theta_s =$ $0'5236rad$	72
Figura 6.21: Simulación para $t_s = 0'75s$ $\theta_s = 0'7854rad$	72



Figura 6.22: Resultados de ángulos para la simulación $t_s = 0'75s$ $\theta_s = 0'7854rad$	73
Figura 6.23: Simulación para $t_s = 0'75s$ $\theta_s = 1'0472rad$	73
Figura 6.24: Resultados de ángulos para la simulación $t_s = 0'75s$ $\theta_s = 1'0472rad$	74
Figura 6.25: Simulación para $t_s = 0'75s$ $\theta_s = 1'309rad$	74
Figura 6.26: Resultados de ángulos para la simulación $t_s = 0'75s$ $\theta_s = 1'309rad$	75
Figura 6.27: Simulación para $t_s = 0'75s$ $\theta_s = 1'5708rad$	75
Figura 6.28: Resultados de ángulos para la simulación $t_s = 0'75s$ $\theta_s = 1'5708rad$	76
Figura 6.29: Representación gráfica de los errores de periodos	78
Figura 6.30: Representación gráfica de los errores de θ	78
Figura 6.31: Representación gráfica de los errores de γ	79
Figura 6.32: Representación gráfica de los errores de distancias	79
Figura 6.33: Simulación para $t_s = 0'75s$ y $\mu=1$	80
Figura 6.34: Simulación para $t_s = 0'75s$ y $\mu=1'5$	80
Figura 6.35: Simulación para $t_s = 1s$ y $\mu=0'5$	81
Figura 6.36: Simulación para $t_s = 1s$ y $\mu=0'9$	81
Figura 6.37: Simulación para $t_s = 1'25s$ y $\mu=0'5$	81
Figura 6.38: Simulación para $t_s = 1'25s$ y $\mu=0'7$	81
Figura 6.39: Creación de la función Tf2	83
Figura 6.40: Creación de la función Th2	83
Figura 6.41: Especificaciones de los pares introducidos	84
Figura 6.42: Simulación doble paso en función del tiempo	85
Figura 6.43: Evolución de ángulos en la simulación de dos pasos	86



Índice de tablas

Tabla 6.1: Resultados de simulaciones con cambios de periodos.....	67
Tabla 6.2: Cálculo de errores en simulaciones con cambio de periodo	67
Tabla 6.3: Resultados de simulaciones con cambios de amplitud	77
Tabla 6.4: Cálculo de errores en simulaciones con cambio de amplitud	77
Tabla 6.5: Cálculo de errores en simulació de dos pasos	87



CAPÍTULO 1

Objetivos

1 OBJETIVOS

El objetivo principal de este proyecto es estudiar la robustez del algoritmo de control propuesto en la referencia [1] para un bípodo simple formado por dos patas actuadas desde un disco. Para ello se realizarán numerosas simulaciones con el programa Adams View del caminar del bípodo variando parámetros de control, como son el periodo y la amplitud de paso, así como el coeficiente de rozamiento entre las patas y el suelo.

Objetivos secundarios que se desprenden de éste son:

- Realizar una revisión del estado del arte de bípodos haciendo énfasis en el estudio y comprensión del artículo “Center of percussion and gait design of biped robots” [1] en el que basa el desarrollo del presente proyecto.
- Describir las partes y componentes del bípodo, sus dimensiones y propiedades mecánicas.
- Modelar con el programa Adams las partes del bípodo y ensamblarlas.
- Estudiar las capacidades de Adams para simular el movimiento e implementar ecuaciones de control.
- Realizar simulaciones con el modelo y el algoritmo de control para diferentes valores del periodo de control y la amplitud de control, registrando la evolución temporal de las variables de posición a lo largo de un paso. Obtener como resultado de dichas simulaciones los valores del periodo y de la amplitud.
- Realizar una comparación entre los valores introducidos en el algoritmo de control y los obtenidos en las correspondientes simulaciones, evaluando las diferencias y estimando así un rango de valores de control para los que el algoritmo es más robusto.
- Realizar simulaciones variando el coeficiente de rozamiento entre las patas y el suelo para diferentes velocidades de paso.
- Realizar simulaciones en Adams de más de un paso del robot, lo que supone intercambiar los pares de control entre las patas (la pata de apoyo pasa a ser la pata en el aire y viceversa). El modelo así definido podrá ser empleado en futuros trabajos.



CAPÍTULO 2

Estado del arte

2 INTRODUCCIÓN

Según la Real Academia Española se define robot como: “máquina o ingenio electrónico programable, capaz de manipular objetos y realizar operaciones antes reservadas solo a personas” [2]. En la práctica, esto es por lo general un sistema electromecánico que, por su apariencia o sus movimientos, ofrece la sensación de tener un propósito propio [3]. La palabra robot puede referirse tanto a mecanismos físicos como a sistemas virtuales de software, aunque suele aludirse a los segundos con el término de bots.

No hay un consenso sobre qué máquinas pueden ser consideradas robots, pero sí existe un acuerdo general entre los expertos y el público sobre que los robots tienden a hacer parte o todo lo que sigue: moverse, hacer funcionar un brazo mecánico, sentir y manipular su entorno y mostrar un comportamiento inteligente, especialmente si éste comportamiento imita al de los humanos o a otros animales.

Aunque hay largas historias sobre ayudantes y acompañantes artificiales, así como los intentos de crearlos, las máquinas totalmente autónomas no aparecieron hasta el siglo XX.

2.1 LOS PRIMEROS AUTÓMATAS

En el siglo IV antes de Cristo, el matemático griego Arquitas de Tarento construyó un ave mecánica que funcionaba con vapor y al que llamó "La paloma". También el ingeniero Herón de Alejandría (10-70 d. C.) creó numerosos dispositivos automáticos que los usuarios podían modificar, y describió máquinas accionadas por presión de aire, vapor y agua. Por su parte, el estudioso chino Su Song levantó una torre de reloj en 1088 con figuras mecánicas que daban las campanadas de las horas.

Al Jazarií (1136–1206), un inventor musulmán de la dinastía Artuqid, diseñó y construyó una serie de máquinas automatizadas, entre los que había útiles de cocina, autómatas musicales que funcionaban con agua, y en 1206 los primeros robots humanoides programables. Las máquinas tenían el aspecto de cuatro músicos a bordo de un bote en un lago, entreteniéndolo a los invitados en las fiestas reales. Su mecanismo tenía un tambor programable con clavijas que chocaban con pequeñas palancas que accionaban instrumentos de percusión. Podían cambiarse los ritmos y patrones que tocaba el tamborilero moviendo las clavijas [3].

El artesano japonés Hisashige Tanaka(1799–1881), conocido como el "Edison japonés", creó una serie de juguetes mecánicos extremadamente complejos, algunos de los cuales servían té, disparaban flechas e incluso trazaban un kanji (carácter japonés).

Por otra parte, desde la generalización del uso de la tecnología en procesos de producción con la Revolución industrial se intentó la construcción de dispositivos automáticos que ayudasen o sustituyesen al hombre. Entre ellos destacaron los Jaquemarts, muñecos de dos o más posiciones que golpean campanas accionados por mecanismos de relojería china y japonesa.

Más tarde se inicio la investigación en robots caminantes, comenzando con pequeños hexápodos y otros tipos de robots de múltiples patas. Estos robots imitaban insectos y artrópodos en funciones y forma.

Con más de cuatro piernas, estos robots son estáticamente estables lo que hace que el trabajar con ellos sea más sencillo. Sólo recientemente se han hecho progresos hacia los robots con locomoción bípeda, que detallaremos a continuación.

2.2 ROBOTS BÍPEDOS

El hombre siempre se ha sentido atraído por la idea de diseñar máquinas semejantes a sí mismo. Estos robots humanoides podrían servir para sustituirnos en tareas molestas o peligrosas, para el cuidado de personas con discapacidad o simplemente como una manera de entretenimiento.

Para explicar los modelos de bípedos con más exactitud es necesaria una primera clasificación:

- Activos: aquellos cuya caminata está directamente relacionada con sistemas complejos de control, así como con la utilización de diversos actuadores y sensores para realizar su cometido, permitiendo así desarrollar tareas complejas.
- Pasivos: aquellos que pueden caminar sin necesidad de ningún tipo de actuador, ya que su movimiento es estimulado únicamente por la acción de la gravedad y por una pequeña pendiente en el terreno.
- Semiactivos: aquellos que combinan las principales características de los dos tipos anteriores.

2.2.1 Robots bípedos pasivos

A principios de los años noventa, Tad McGeer demostró que un mecanismo con la cinemática y dinámica adecuadas podía bajar una pendiente suave sin ningún aporte externo de energía. McGeer bautizó a estos bípedos como “Robots Pasivos Dinámicos” [4].

Los prototipos desarrollados por McGeer tenían unas características que le hacían muy diferentes a los robots humanoides tradicionales. La particularidad más importante, es que tenían los pies curvos, lo que impedía que fueran controlables, ignorando la necesidad de estabilidad local.

Cabe destacar también el bípedo tridimensional de la Universidad de Cornell, ya que fue el primer “Passive Dynamic Walker” que no tenía dos pares de piernas, empleaba sus brazos para lograr la estabilidad lateral.

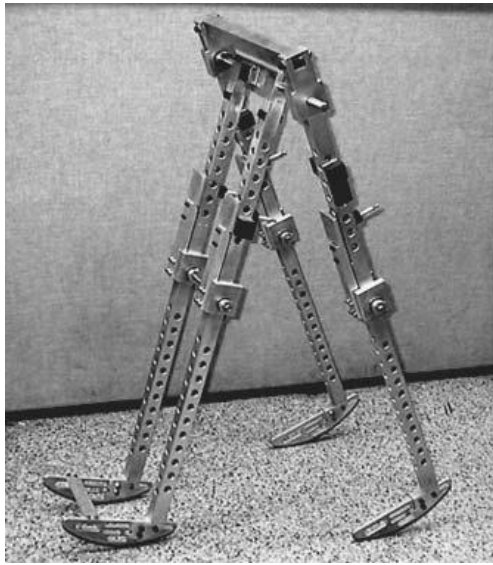


Figura2.1 Dynamite “Passive Dynamic Walker” con rodillas de McGeer y bípedo tridimensional de la Universidad de Cornell.

Comenzó a explorarse más tarde la posibilidad de sustituir la energía que proporcionaba la pendiente por algún tipo de actuador, surgiendo así los llamados Robots de Ciclo Límite entre los que destacamos los tres modelos que se dieron a conocer en la reunión anual de la *Asociación Americana para el Avance de la Ciencia* (AAAS) de 2005, cuyos pasos y movimientos se parecían a la forma de andar de los humanos:

- Ranger, el bípedo eficiente de Cornell: su principal característica es que fue diseñado para lograr la máxima eficiencia energética, consiguiendo ser la única máquina andante no pasiva que consume menos energía al andar que el hombre. Compensa la energía que pierde al andar dando un impulso en el tobillo de la pierna de apoyo justo cuando la pierna en oscilación colisiona con el suelo. Sólo es capaz de caminar hacia delante.



Figura 2.2 Robot Ranger de la Universidad de Cornell

- Toddler del MIT: es el más sofisticado de los robots de Ciclo Límite ya que puede caminar en otras direcciones. Incluye un novedoso sistema de aprendizaje que le permite adaptarse de forma continuada al terreno sobre el que se mueve, se convierte así en el primero que puede andar sin tener información previamente implantada en sus controles.

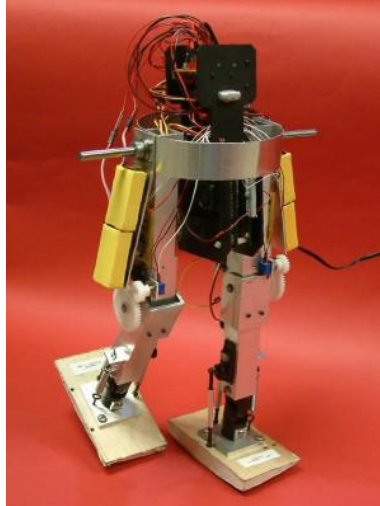


Figura 2.3 Robot Toddler del MIT

- Denise de la Universidad de Delft: mediante un diseño adecuado de la articulación del tobillo se acopla la inclinación lateral del robot con el giro respecto a su eje vertical, lográndose una estabilidad lateral suficiente como para andar. Para ello, la articulación del tobillo permite el giro en la dirección de la marcha. La cadera está equipada con músculos Mckibben, unos actuadores neumáticos que pretenden reproducir los músculos humanos.



Figura 2.4 Robot Denise de la Universidad de Delft

En resumen, estos tres robots tienen una serie de características comunes. En primer lugar, fijándonos en su diseño mecánico, todos tienen pocos grados de libertad. Respecto a las estrategias de control todas son muy sencillas, ya que incluso en Toddler el aprendizaje reforzado se usa para obtener un número reducido de parámetros. Además la cantidad de información sensorial necesaria para el control de la marcha es muy reducida.

2.2.2 Robots bípedos activos

Si nos centramos en los robots bípedos que se desarrollaron más tarde, los más impresionantes son los basados en la técnica denominada “*Zero Moment Point*”, o “*Punto de Momento Cero*”. Esta técnica de control está basada en la especificación del punto con respecto al cuál la fuerza de reacción dinámica en el contacto del pie con el suelo no produce ningún momento, es decir el punto en que la inercia total de la fuerza es igual a cero. El concepto asume el área de contacto es plana y que tiene una fricción suficientemente alta como para no producir deslizamiento en los pies.

Se han propuesto otros criterios de estabilidad como el “Indicador de la Rotación del Pie”, aunque la mayoría de los robots humanoides utilizan el ZMP. Sea cual sea el criterio de estabilidad local utilizado, una vez garantizada ésta, el robot es capaz de caminar de forma segura.

Basándose en estas ideas los robots humanoides han logrado realizar una gran variedad de tareas, como subir o bajar escaleras, evitar obstáculos o manipular objetos. Sin embargo, todavía no se ha resuelto el problema de caminar en entornos no estructurados, sobre superficies irregulares o bajo la influencia de perturbaciones externas, situaciones que el ser humanos maneja con relativa facilidad.

El primer robot antropomórfico capaz de caminar fue el Wabot-1 de la Universidad de Waseda. A partir de entonces diversos grupos de investigación han desarrollado otros modelos, la mayoría en Japón, pero también en otros países, incluyendo España. A continuación se describen algunos de estos modelos

- Wabot-1, Universidad de Waseda [5]: además de la capacidad de andar podemos destacar su capacidad de comunicarse con una persona en japonés, poder medir distancias y direcciones a los objetos y agarrar y transportar objetos con las manos.

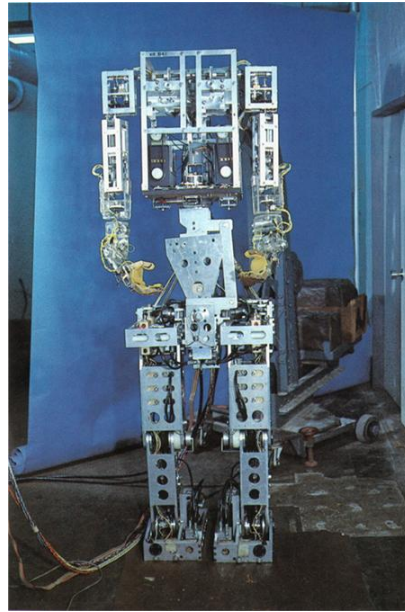


Figura 2.5 Wabot-1 de la Universidad de Waseda

- Wabot-2, Universidad de Waseda: incorpora la capacidad de leer una partitura musical, de tocar un órgano electrónico y de acompañar a una persona mientras la escucha cantar.

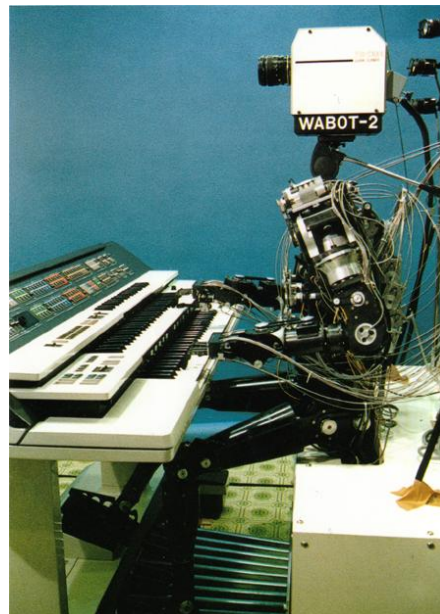


Figura 2.6 Wabot-2 de la Universidad de Waseda

- Asimo de Honda: los modelos de Honda han ido evolucionando desde modelos como el E0, P2 o P3 hasta el Asimo [7]. Entre sus habilidades podemos destacar la capacidad para interactuar con los humanos; su capacidad de reconocimiento facial, así como de gestos y movimientos;

puede correr a 6 km/h y en su última versión trabaja en equipo con otros robots como él y puede moverse cómodamente entre personas.

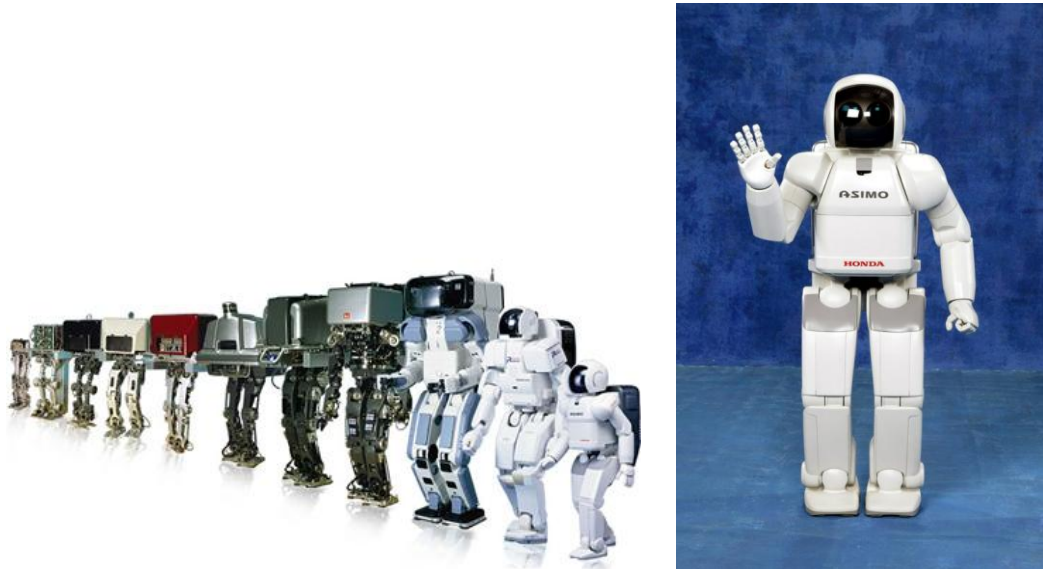


Figura 2.7 Evolución de los robots de Honda y Asimo

- QRIO de Sony: este modelo desarrollado por Sony tiene sentido del equilibrio, que le permite andar por superficies irregulares y rugosas, ponerse de pie tras una caída, además de las capacidades de reconocer voces y su procedencia, calcular distancias y posición de objetos y de mantener conversaciones utilizando el lenguaje corporal para expresarse.



Figura 2.8 QRIO de Sony

- HRP-2P, Universidad de Kawada y AIST: Tiene 30 grados de libertad y el diseño de sus caderas y entrepierna le permite caminar en lugares estrechos. Se ha empleado en su diseño electrónica de alta densidad, lo que permite que en una pequeña "mochila" se integren los ordenadores, electrónica y baterías necesarios para su funcionamiento.

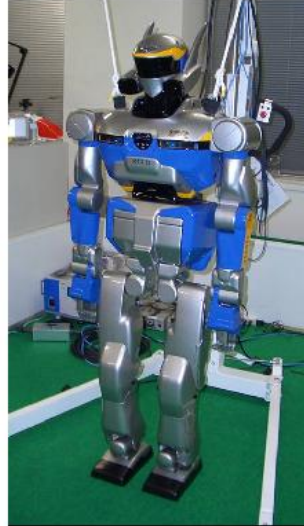


Figura 2.9 HPR-2P

- Wabian 2, Universidad de Waseda [8]: el robot cuenta con siete grados de libertad en sus piernas, permitiendo que cualquier cambio de dirección de éstas no esté restringido por la posición de su pelvis. Incorpora además un novedoso sistema de pies con un dedo pasivo que le permite realizar pasos largos imitando al ser humano, ya que el talón y los dedos ya no llegan al suelo al mismo tiempo.



Figura 2.10 Wabian 2 de la Universidad de Waseda.

Uno de los mayores inconvenientes de los robots mencionados es su peso y su consumo. Si se consiguen diseñar nuevos mecanismos y cadenas cinemáticas, manteniendo el mismo número de grados de libertad que un robot humanoide, pero con menor número de actuadores y reductores, es posible reducir drásticamente el peso y el coste del robot. Por supuesto, asociado a este hecho, está la disminución del consumo de energía del robot, ya que la mayor parte de ésta es requerida por los motores. Esta mejora conllevaría una disminución de sus baterías, con la consiguiente disminución también del peso total del robot.

CAPÍTULO 3

Diseño del robot

3 DISEÑO DEL ROBOT

3.1 DESCRIPCIÓN DEL MOVIMIENTO

El diseño del robot tiene como principal característica la simplicidad de sus elementos, con el fin de disminuir el número de actuadores y el peso, reduciendo por lo tanto su consumo.

El caminar estable del bípedo se pretende conseguir mediante un algoritmo de control en el que el periodo y la longitud de paso son parámetros a definir por el usuario.

El bípedo está compuesto por dos patas, que entran en contacto con el suelo de forma alternante. Cada pata se une de forma independiente pero concéntrica a un disco situado en la parte central mediante un eje. No se dota de rodillas a las patas, con lo que se reducen los grados de libertad del conjunto total, simplificando las ecuaciones de control que se desarrollan en apartados posteriores.

El movimiento se basa en la generación de un péndulo invertido al girar el conjunto respecto al Zero Moment Point (ZMP), situado en el punto de contacto existente entre la pata apoyada y la superficie de apoyo. Para que el conjunto gire respecto a este punto será necesario aplicar un par que implique que el resto del conjunto del robot efectúe el movimiento de avance. Ante la dificultad de aplicar el par directamente sobre este punto y dado que cualquier par aplicado sobre el robot se ve reflejado sobre el ZMP, dicho par será aplicado entre el disco central y las patas.

Por otro lado, la pata que no está apoyada realizará una rotación respecto al eje, avanzando hasta que entre en contacto con el suelo en una posición más avanzada que el resto del conjunto, convirtiéndose en la nueva pata de apoyo. Se puede por lo tanto dividir el movimiento en dos fases, una primera en la que una única pata está en contacto con el suelo; y una segunda en la que las dos patas se encuentran apoyadas antes de realizarse el cambio de una pata a otra.

Serán necesarios por lo tanto dos controladores, vinculados uno a cada pata, que controlarán de forma independiente el conjunto (considerado como un péndulo invertido) y la pata en el aire.

Los dos controladores inician el movimiento, el robot comienza a girar respecto al ZMP, mientras que la pata situada en la posición más atrasada se despega del suelo y comienza un movimiento pendular hasta que entra en contacto de nuevo con la superficie de apoyo. En este momento se produce el intercambio entre los controladores, convirtiéndose el controlador antes encargado del cuerpo del robot en el de la pata en el aire y viceversa.

Realizando este procedimiento de manera repetida se consigue que el robot “camine”.

Uno de los principales problemas que aparecen a la hora de efectuar el movimiento, dejando a un lado la parte referida a las ecuaciones de control, son las posibles interacciones no deseadas que se producen entre las piezas y el suelo al no haber dotado de rodillas a nuestro modelo.

Como concepto previo es necesario conocer las tres posibles posiciones en las que se pueden cruzar ambas patas a lo largo del paso. Las patas pueden encontrarse antes de llegar al eje vertical, en el propio eje, o pasado este. (Ver figura 3.1)

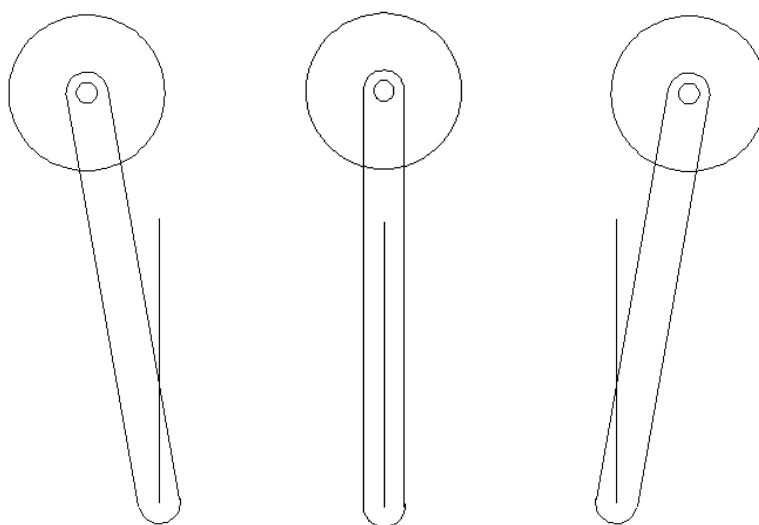


Figura 3.1 Posiciones de cruce entre patas.

Se pueden utilizar diferentes soluciones para este problema, pero la que a priori parece más adecuada por su fácil aplicación y porque no sería necesaria la modificación física del modelo ni de las ecuaciones que lo controlan, sería realizar el movimiento de avance sobre unas plataformas, de manera que estas tengan el tamaño necesario para que las pata se apoyen pero que no interfieran con el resto del robot. Se tratará este tema de manera más detallada en los posteriores apartados de modelado y experimentación.

3.2 DISEÑO DEL ROBOT

Una vez definidas las características generales que debe cumplir el bípido se puede detallar las características concretas de cada pieza. La simplicidad del modelo conlleva la elección de formas sencillas para los componentes que lo forman, siendo el número de piezas el menor posible.

Las formas, así como sus medidas acotadas se describen a continuación en las figuras 3.2 y 3.3.

Actuadores

Los actuadores no se incluyen como elementos físicos en este modelo, pero se encontrarían fijados uno a cada lado del disco, con el extremo contrario en cada pata.

Patas

Forma prismática con los extremos formando semicircunferencias, de 420 mm de alto, 20 mm de ancho y 30 mm de largo.

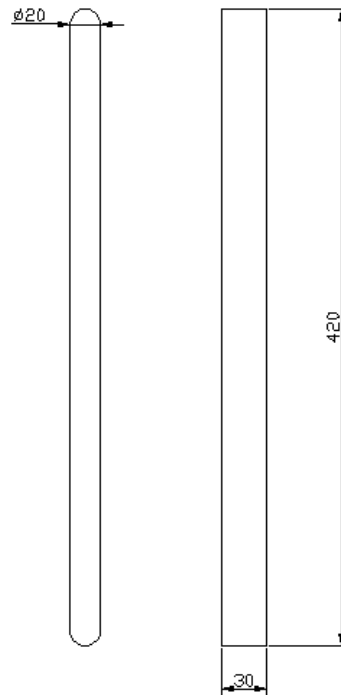


Figura 3.2 Medidas de pata del modelo.

Disco

El disco encargado de generar el movimiento tiene un diámetro de 75 mm y una anchura de 30 mm.

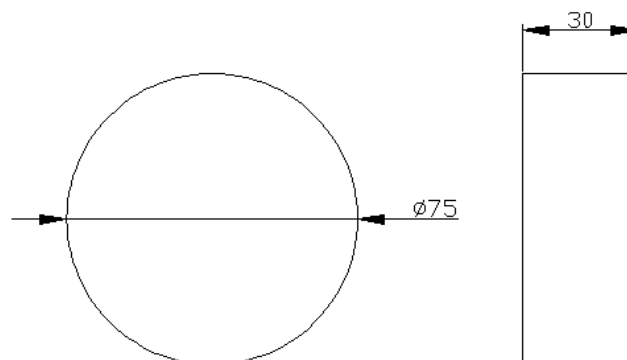


Figura 3.3 Medidas del disco del modelo.

CAPÍTULO 4

Estudio teórico del péndulo invertido

4 ESTUDIO TEÓRICO

El análisis teórico del modelo propuesto tiene como objetivo la adquisición de las herramientas matemáticas que permitan controlar el movimiento del bípodo.

Estas ecuaciones actúan sobre el par generado por cada motor sobre el disco, efectuando el movimiento con la longitud de paso y periodo que se desee. Se consigue por lo tanto una herramienta con la que efectuar infinitas combinaciones de movimientos siempre que las dimensiones constructivas lo permitan.

4.1 CARACTERÍSTICAS GEOMÉTRICAS

Antes de realizar los diferentes estudios teóricos se define en la figura 4.1 la nomenclatura a seguir para los diferentes ángulos que definen la posición de las piezas.

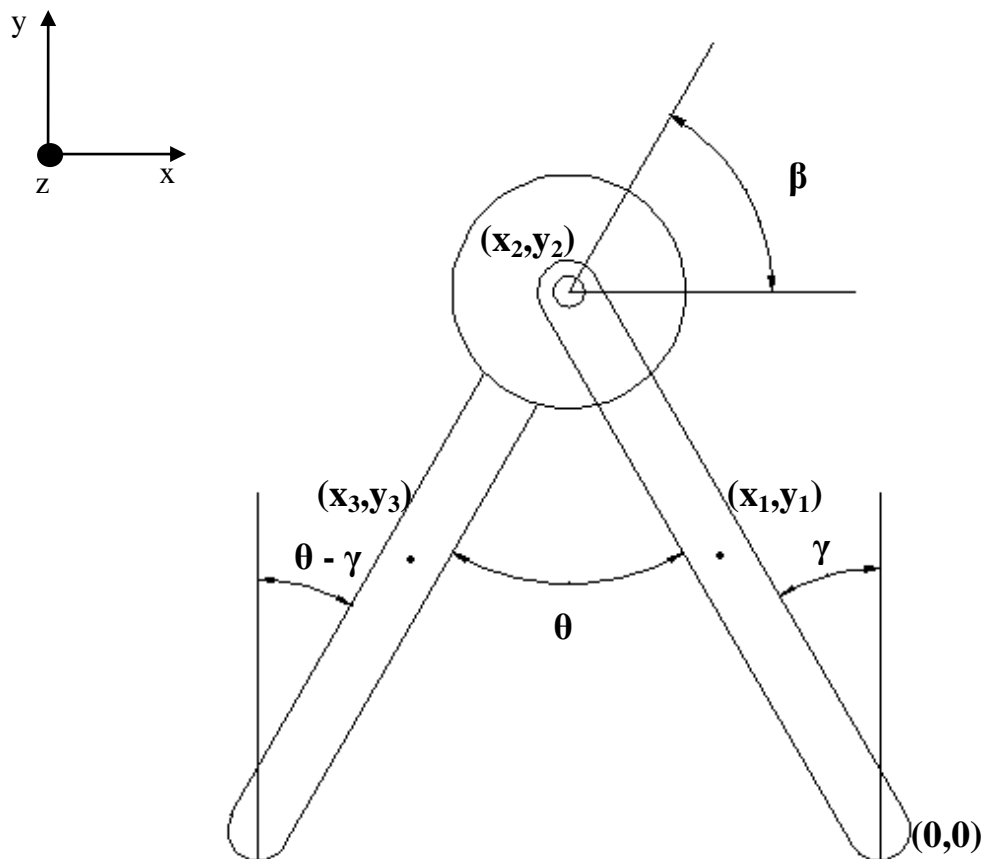


Figura 4.1 Descripción de ángulos del modelo.

Las longitudes totales de las patas se denominan mediante una l .

4.2 ESTUDIO CINEMÁTICO

Se pueden expresar las posiciones de los centros de gravedad en función de los valores que toman en función del tiempo los ángulos y las longitudes.

$$x_1 = -\frac{l}{2} \sin \gamma;$$

$$y_1 = \frac{l}{2} \cos \gamma;$$

$$x_2 = -l \sin \gamma;$$

$$y_2 = l \cos \gamma$$

$$x_3 = -l \sin \gamma - \frac{l}{2} \sin(\theta - \gamma);$$

$$y_3 = l \cos \gamma - \frac{l}{2} \cos(\theta - \gamma)$$

Si se derivan las ecuaciones anteriormente obtenidas, se consiguen las expresiones para la velocidad:

$$\dot{x}_1 = -\frac{l}{2} \dot{\gamma} \cos \gamma$$

$$\dot{y}_1 = -\frac{l}{2} \dot{\gamma} \sin \gamma$$

$$\dot{x}_2 = -l \dot{\gamma} \cos \gamma$$

$$\dot{y}_2 = -l \dot{\gamma} \sin \gamma$$

$$\dot{x}_3 = -l \dot{\gamma} \cos \gamma - \frac{l}{2} (\dot{\theta} - \dot{\gamma}) \cos(\theta - \gamma)$$

$$\dot{y}_3 = -l \dot{\gamma} \sin \gamma + \frac{l}{2} (\dot{\theta} - \dot{\gamma}) \sin(\theta - \gamma)$$

Y con una segunda derivación se obtienen las aceleraciones:

$$\ddot{x}_1 = -\frac{l}{2} \ddot{\gamma} \cos \gamma + \frac{l}{2} \dot{\gamma}^2 \sin \gamma$$

$$\ddot{y}_1 = -\frac{l}{2} \ddot{\gamma} \sin \gamma - \frac{l}{2} \dot{\gamma}^2 \cos \gamma$$

$$\ddot{x}_2 = -l \ddot{\gamma} \cos \gamma + l \dot{\gamma}^2 \sin \gamma$$

$$\ddot{y}_2 = -l \ddot{\gamma} \sin \gamma - l \dot{\gamma}^2 \cos \gamma$$

$$\ddot{x}_3 = -l \ddot{\gamma} \cos \gamma + l \dot{\gamma}^2 \sin \gamma - \frac{l}{2} (\ddot{\theta} - \ddot{\gamma}) \cos(\theta - \gamma) + \frac{l}{2} (\dot{\theta} - \dot{\gamma})^2 \sin(\theta - \gamma)$$

$$\ddot{y}_3 = -l \ddot{\gamma} \sin \gamma - l \dot{\gamma}^2 \cos \gamma + \frac{l}{2} (\ddot{\theta} - \ddot{\gamma}) \sin(\theta - \gamma) + \frac{l}{2} (\dot{\theta} - \dot{\gamma})^2 \cos(\theta - \gamma)$$

4.3 ESTUDIO DINÁMICO

Una vez halladas las ecuaciones que describen la cinemática del conjunto, se comienza el estudio de las ecuaciones dinámicas. Para ello dividimos el conjunto en las piezas que lo forman, teniendo en cuenta las diferentes reacciones y momentos que se producen.

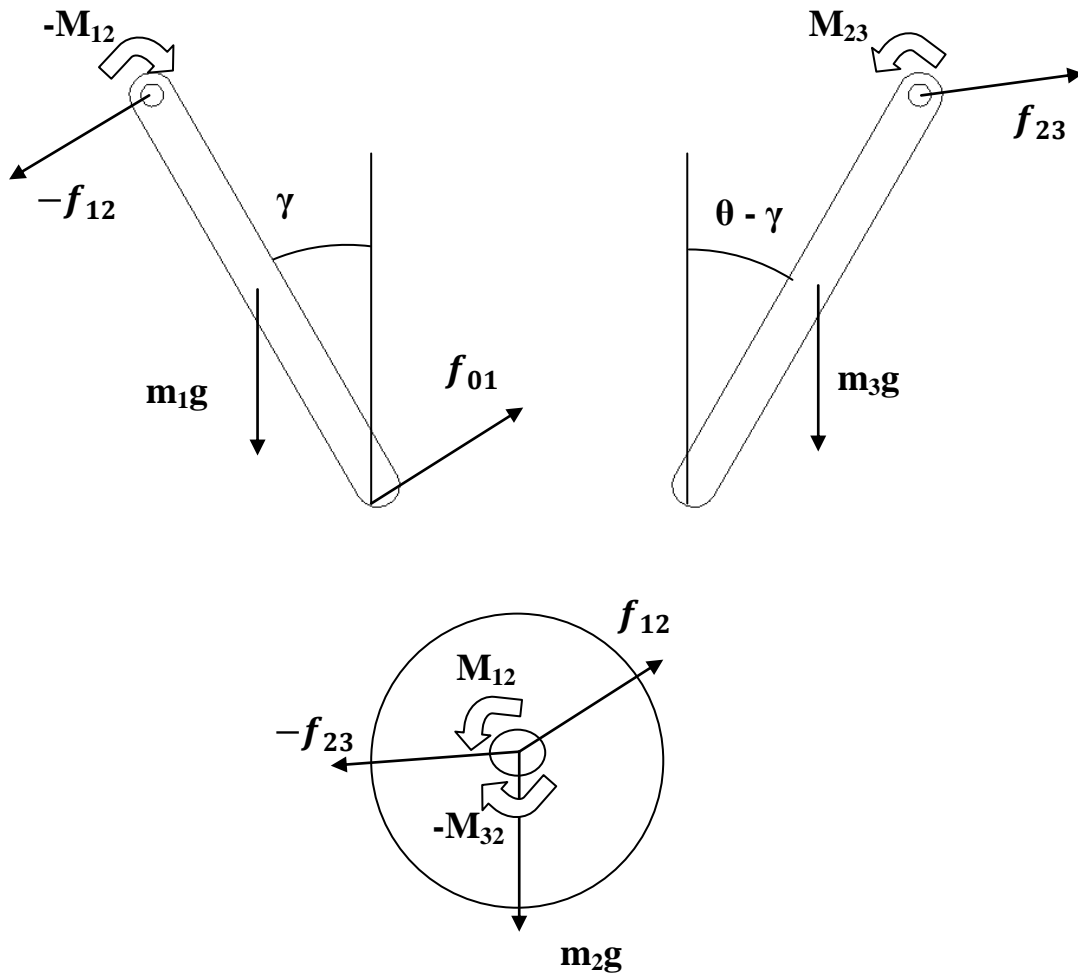


Figura 4.2 Reacciones y momentos en las piezas.

Si se realiza: $\sum F_{horizontales} = 0$; $\sum F_{verticales} = 0$ y $\sum M = I \alpha$, siendo α la aceleración angular, en cada pieza, obtenemos:

Pata apoyada

$$f_{01x} - f_{12x} - m_1 \ddot{x}_1 = 0$$

$$f_{01y} - f_{12y} - m_1 g - m_1 \ddot{y}_1 = 0$$

$$-M_{12} + m_1 g \frac{l}{2} \sin \gamma + l \cos \gamma f_{12x} + l \sin \gamma f_{12y} - I_{1A} \ddot{\gamma} = 0$$

Disco

$$f_{12x} - f_{23x} - m_2 \ddot{x}_2 = 0$$

$$f_{12y} - f_{23y} - m_2 g - m_2 \ddot{y}_2 = 0$$

$$M_{12} - M_{23} - I_{2C} \ddot{\beta} = 0$$

Pata en el aire

$$f_{23x} - m_3 \ddot{x}_3 = 0$$

$$f_{23y} - m_3 g - m_3 \ddot{y}_3 = 0$$

$$M_{23} + \frac{l}{2} \sin(\theta - \gamma) f_{23y} - \frac{l}{2} \cos(\theta - \gamma) f_{23x} - I_{3C} (\ddot{\theta} - \ddot{\gamma}) = 0$$

Si se sustituyen los valores de las aceleraciones calculados en la parte cinemática se pueden obtener las correspondientes ecuaciones dinámicas para cada pieza en función de los ángulos y sus derivadas:

Pata apoyada

$$f_{01x} - f_{12x} + m_1 \frac{l}{2} \ddot{\gamma} \cos \gamma - m_1 \frac{l}{2} \dot{\gamma}^2 \sin \gamma = 0$$

$$f_{01y} - f_{12y} - m_1 g + m_1 \frac{l}{2} \ddot{\gamma} \sin \gamma + m_1 \frac{l}{2} \dot{\gamma}^2 \cos \gamma = 0$$

$$-M_{12} + m_1 g \frac{l}{2} \sin \gamma + l \cos \gamma f_{12x} + l \sin \gamma f_{12y} - I_{1A} \ddot{\gamma} = 0$$

Disco

$$f_{12x} - f_{23x} + m_2 l \ddot{\gamma} \cos \gamma - m_2 l \dot{\gamma}^2 \sin \gamma = 0$$

$$f_{12y} - f_{23y} - m_2 g + m_2 l \ddot{\gamma} \sin \gamma + m_2 l \dot{\gamma}^2 \cos \gamma = 0$$

$$M_{12} - M_{23} - I_{2C} \ddot{\beta} = 0$$

Pata en el aire

$$f_{23x} + m_3 l \ddot{\gamma} \cos \gamma - m_3 l \dot{\gamma}^2 \sin \gamma + m_3 \frac{l}{2} (\ddot{\theta} - \ddot{\gamma}) \cos(\theta - \gamma) - m_3 \frac{l}{2} (\dot{\theta} - \dot{\gamma})^2 \sin(\theta - \gamma) = 0$$

$$f_{23y} - m_3 g + m_3 l \ddot{\gamma} \sin \gamma + m_3 l \dot{\gamma}^2 \cos \gamma - m_3 \frac{l}{2} (\ddot{\theta} - \ddot{\gamma}) \sin(\theta - \gamma) - m_3 \frac{l}{2} (\dot{\theta} - \dot{\gamma})^2 \cos(\theta - \gamma) = 0$$

$$M_{23} + \frac{l}{2} \sin(\theta - \gamma) f_{23y} - \frac{l}{2} \cos(\theta - \gamma) f_{23x} - I_{3C} (\ddot{\theta} - \ddot{\gamma}) = 0$$

Si la cinemática del robot está preestablecida, es decir, se conocen las funciones $\gamma = \gamma(t)$ y $\theta = \theta(t)$, las nueve ecuaciones precedentes forman un sistema lineal para las incógnitas: $f_{01x}, f_{01y}, f_{12x}, f_{12y}, f_{23x}, f_{23y}, M_{12}, M_{23}, \ddot{\beta}$. Dicho sistema se expresa matricialmente de la siguiente manera.

$$\begin{pmatrix} 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & l \sin \gamma & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & l \cos \gamma & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & -I_{2C} \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\frac{l}{2} \cos(\theta - \gamma) & \frac{l}{2} \sin(\theta - \gamma) & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} f_{01x} \\ f_{01y} \\ f_{12x} \\ f_{12y} \\ f_{23x} \\ f_{23y} \\ M_{12} \\ M_{23} \\ \ddot{\beta} \end{pmatrix} = \begin{pmatrix} m_1 \frac{l}{2} \sin \gamma \dot{\gamma}^2 - m_1 \frac{l}{2} \cos \gamma \ddot{\gamma} \\ m_1 g - m_1 \frac{l}{2} \cos \gamma \dot{\gamma}^2 - m_1 \frac{l}{2} \sin \gamma \ddot{\gamma} \\ I_{1A} \ddot{\gamma} - m_1 g \frac{l}{2} \sin \gamma \\ m_2 l \sin \gamma \dot{\gamma}^2 - m_2 l \cos \gamma \ddot{\gamma} \\ m_2 g - m_2 l \cos \gamma \dot{\gamma}^2 - m_2 l \sin \gamma \ddot{\gamma} \\ 0 \\ m_3 l \dot{\gamma}^2 \sin \gamma - m_3 l \ddot{\gamma} \cos \gamma + m_3 \frac{l}{2} (\dot{\theta} - \dot{\gamma})^2 \sin(\theta - \gamma) - m_3 \frac{l}{2} (\ddot{\theta} - \ddot{\gamma}) \cos(\theta - \gamma) \\ m_3 g - m_3 l \dot{\gamma}^2 \cos \gamma - m_3 l \ddot{\gamma} \sin \gamma + m_3 \frac{l}{2} (\dot{\theta} - \dot{\gamma})^2 \cos(\theta - \gamma) + m_3 \frac{l}{2} (\ddot{\theta} - \ddot{\gamma}) \sin(\theta - \gamma) \\ I_{3C} (\ddot{\theta} - \ddot{\gamma}) \end{pmatrix}$$

Este proyecto, sin embargo, tiene por objeto comprobar el algoritmo de control propuesto en [1] mediante simulaciones con Adams, por lo que el desarrollo analítico de la dinámica del robot quede como propuesta de trabajo futuro.

A continuación se define el algoritmo de control.

4.4 ECUACIONES DE CONTROL

El paso se puede dividir en dos partes. La primera es aquella en la que una de las patas está en el aire y la otra en contacto con el suelo, la otra parte del paso es aquella en la que ambas patas están en contacto con la superficie.

Son necesarios dos algoritmos de control, uno para la pata que está en el aire y el otro para controlar el resto del robot. Cuando el paso se complete, la pata que hasta ahora era la que estaba apoyada en el suelo pasa a ser la que está en el aire y viceversa.

Las ecuaciones de control se obtienen del artículo publicado en prensa que se proporciona como adjunto a este proyecto [1].

Nota: si se realiza un análisis dimensional de las ecuaciones proporcionadas en el artículo, se detecta que en éstas existen errores. Se trata de un error tipográfico, en los términos k_p , la parte que se encuentra elevada al cuadrado debe en realidad estar elevada a -2, es decir, debe ser su propio valor inverso. Solucionado este problema se pueden expresar las ecuaciones con su forma correcta.

4.4.1 Pata en el aire

El tiempo que debe emplear la pierna en movimiento en alcanzar la posición final en el paso debe ser menor o igual que el periodo total elegido para paso, para asegurar que está situada en su posición cuando debe actuar como la pata fija. Tenemos por lo tanto:

$$t_f = ct_s, \text{ donde } c \text{ adquiere un valor entre } 0 \text{ y } 1$$

Como se puede ver en la ref. [1] el algoritmo de control para la pata en el aire persigue un movimiento críticamente amortiguado, con lo que la ecuación del par empleada para controlar esta pata es:

$$\tau_f = k_p(\theta - \theta_s) + k_d\dot{\theta} - m_{leg}[\ddot{r}_{leg} \times \vec{g}], \text{ donde:}$$

$$k_p = I_{leg} \left(\frac{ct_s}{7} \right)^{-2} \text{ y } k_d = 2\sqrt{k_p I_{leg}}$$

4.4.2 Pata apoyada

El algoritmo propuesto también pretende un movimiento de oscilador críticamente amortiguado para el péndulo invertido al que puede reducirse el bípodo; lo que corresponde a la siguiente ecuación de control:

$$\tau_h = k_{po} \left(\gamma - \frac{\theta_s}{2} \right) + k_{do} \dot{\gamma} - m_{robot} |\vec{r}_{robot} \times \vec{g}| - \tau_f, \text{ donde:}$$

$$k_{po} = I_{robot} \left(\frac{t_s}{7} \right)^{-2} \text{ y } k_{do} = 2\sqrt{k_{po} I_{robot}}$$

4.5 CÁLCULO CONSTANTES Y VARIABLES

Una vez dispuestas las diferentes ecuaciones de control, se procede al cálculo de las diferentes constantes y variables que intervienen en éstas, teniendo en cuenta las dimensiones especificadas en el apartado de diseño del bípodo.

Parte de estos datos serán proporcionados por la herramienta informática de diseño en el momento de la simulación, pero es importante conocer la procedencia de los cálculos, teniendo utilidad además en el caso de que se quieran implementar las ecuaciones de control sin necesidad de disponer del programa.

4.5.1 Cálculo de las masas de los componentes

Las medidas de las piezas aparecen en las figuras 3.2 y 3.3, y el material escogido para las piezas es acero con densidad $7801 \frac{kg}{mm^3}$

Pata

El volumen y la masa de las patas tratando cada una como un conjunto de un paralelepípedo y dos semicilindros es:

$$V_{leg} = A_{base} h + \pi r_1^2 e = 249424'778 \text{ mm}^3$$

$$m_{leg} = V_{leg} \rho_{acero} = 1'94576 kg$$

A_{base}	20 mm*30 mm=600 mm ²
h	400 mm
r_1	10 mm
e	30 mm

Conjunto robot

Para el cálculo de la masa del conjunto robot se tienen en cuenta las masas de las diferentes piezas:

$$V_{disco} = \pi R^2 e = 132535'94 \text{ mm}^3$$

$$m_{disco} = V_{disco} \rho_{acero} = 1'03391 kg$$

R	37'5 mm
e	30 mm

$$m_{robot} = 2m_{leg} + m_{disco} = 4'92543kg$$

Resumiendo tenemos:

$$m_{leg} = 1'94576kg$$

$$m_{robot} = 4'92543kg$$

4.5.2 Cálculo de los centros de masas de los componentes

La aplicación de las leyes de control expuestas exige el conocimiento de los centros de masa de las partes del bípodo. El centro de gravedad de las piezas individuales está situado en el centro geométrico de estas.

En la ecuación se efectúa un producto vectorial entre la posición del centro de gravedad y la aceleración de la gravedad, por lo tanto se divide esta posición del centro de gravedad en sus diferentes componentes. Si se realiza dicho producto vectorial se puede observar que las únicas componentes necesarias serán la x y z.

$$\vec{r} \times \vec{g} = \begin{vmatrix} \vec{i} & \vec{j} & \vec{k} \\ r_x & r_y & r_z \\ 0 & g & 0 \end{vmatrix} = -r_z g \vec{i} + r_x g \vec{k}$$

Además según nuestro sistema de referencia y las ecuaciones descritas, los pares solo actúan en el eje z, por lo que el verdadero valor del producto vectorial que influye es el asociado a \vec{k} .

Para facilitar la comprensión de las medidas, se describen en la figura 4.3:

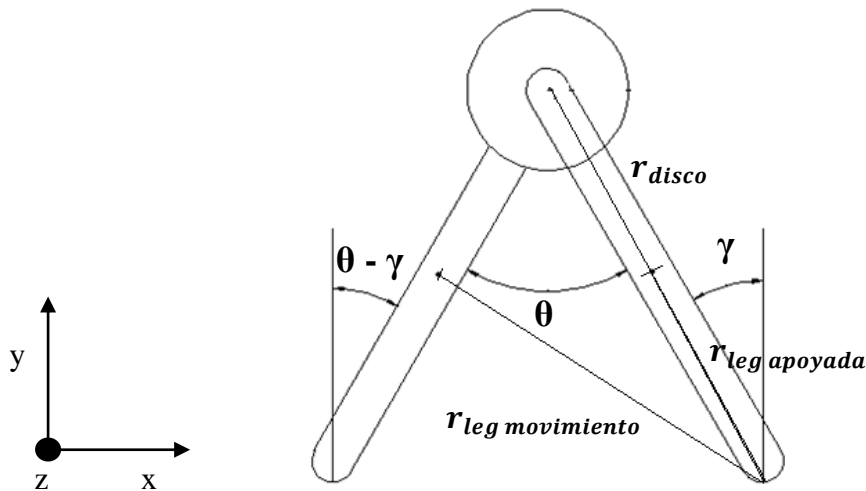


Figura 4.3 Descripción de distancias del modelo.

Pata en el aire

En este caso la posición del centro de gravedad se referencia respecto al eje sobre el que realiza la rotación, es decir, aquel que une a las dos patas y el disco. Por lo tanto teniendo en cuenta donde se encuentra el centro de gravedad y las diferentes medidas de las piezas se tiene:

$$r_{leg\ x} = 200 \sin(\theta - \gamma)$$

Conjunto robot

El conjunto total se referencia respecto al ZMP (Zero Moment Point) situado en el punto de contacto de la pata de apoyo. Al tratarse de diferentes masas en movimiento se expresa primero cada centro de gravedad de forma individual.

$$r_{leg\ apoyada\ x} = 210 \sin \gamma$$

$$r_{disco\ x} = 410 \sin \gamma$$

$$r_{leg\ movimiento\ x} = 410 \sin \gamma + 200 \sin(\theta - \gamma)$$

La distancia por lo tanto del centro de gravedad total del conjunto referenciado respecto al ZMP se puede calcular como:

$$r_{robot\ x} = \frac{m_{leg} r_{leg\ apoyada\ x} + m_{disco} r_{disco\ x} + m_{leg} r_{leg\ movimiento\ x}}{m_{robot}}$$

$$r_{robot\ x} = 330'99 \sin \gamma + 79'01 \sin(\theta - \gamma)$$

4.5.3 Cálculo de inercias de los componentes

Pata

Se calcula en primer lugar la inercia de la pata en su centro de gravedad. Teniendo en cuenta la forma de esta, se puede dividir en diversas piezas.

$$I_{paralelepipedo} = \frac{m_p}{12} (b^2 + c^2)$$

$$I_{cilindro} = \frac{1}{2} m_c r^2$$

$$I_{cilindro\ descentrado} = m_c d^2 = m_c \left(\frac{c}{2} + \frac{4r}{3\pi} \right)^2$$

m_p	1'87226 kg
b	20 mm
c	400 mm
e	30 mm
m_c	0'0735 kg
r	10 mm

$$I_{leg} = I_{paralelepipedo} + I_{cilindro} + I_{cilindro\ descentrado} = 28095'65273 kgmm^2$$

Esta es la inercia de la pieza respecto de su centro de masas, pero para emplearla en la ecuación debe estar referenciado respecto del punto de unión con el eje.

$$I'_{leg} = I_{leg} + md^2 = I_{leg} + m_{leg} \left(\frac{c}{2}\right)^2$$

$$I'_{leg} = 105926'0527 kgmm^2$$

Conjunto robot

El conjunto robot estará formado por la pata apoyada, el disco central y la pata en movimiento.

En primer lugar se calcula la pata apoyada, utilizando el cálculo del apartado anterior pero referenciándolo desde el ZMP

$$I_{leg\ apoyada} = I_{leg} + m_{leg} \left(\frac{c}{2} + r\right)^2 = 113903'6687 kgmm^2$$

La inercia del disco también se referencia desde el ZMP y tendrá un valor constante al ser la distancia desde este punto al centro de masas siempre la misma.

$$I_{disco} = \frac{1}{2} m_d R^2 + m_d (c + r)^2 = \frac{1}{2} 1'03391 \cdot 37'5^2 + 1'03391 (400 + 10)^2 = 174527'239 kgmm^2$$

m _d	1'03391 kg
R	37'5 mm

En el caso de la pata en movimiento la distancia entre el ZMP y centro de gravedad varía en función de los ángulos, por lo que se toma la medida como la resultante de sus componentes x e y, que aparecen en el estudio cinemático realizado con anterioridad en el punto 4.2.

$$I_{leg\ movimiento} = I_{leg} + m_{leg} d^2 = I_{leg} + m_{leg} ((-(c + r) \sin \gamma - \left(\frac{c}{2} + r \sin \theta - \gamma\right)^2 + (c + r \cos \gamma - c^2 + r \cos \theta - \gamma)^2) = 28095'65273 + 1'94576 ((-410 \sin \gamma - 210 \sin(\theta - \gamma))^2 + (410 \cos \gamma - 210 \cos(\theta - \gamma))^2)$$

Se puede obtener por lo tanto la inercia total del conjunto como la suma de las anteriores.

$$I_{robot} = I_{leg\ apoyada} + I_{disco} + I_{leg\ movimiento}$$

$$I_{robot} = 316526'5604 + 1'94576 ((-410 \sin \gamma - 210 \sin(\theta - \gamma))^2 + (410 \cos \gamma - 210 \cos(\theta - \gamma))^2)$$

4.6 ECUACIONES DE CONTROL FINALES

Obtenidos los diferentes parámetros se está en condiciones de desarrollar las ecuaciones de control del par, obteniendo las aplicables a nuestro modelo. Se introducen además los términos k_p y k_d , simplificando la ecuación.

4.6.1 Pata en movimiento

$$\tau_f = k_p(\theta - \theta_s) + k_d\dot{\theta} - m_{leg}|\vec{r}_{leg} \times \vec{g}| = I_{leg} \left(\frac{c t_s}{7}\right)^{-2} (\theta - \theta_s) + 2 \sqrt{I_{leg}^2 \left(\frac{c t_s}{7}\right)^{-2}} \dot{\theta} - m_{leg} g r_{leg x}$$

$$\tau_f = I_{leg} \left(\frac{c t_s}{7}\right)^{-2} (\theta - \theta_s) + 2 I_{leg} \left(\frac{c t_s}{7}\right)^{-1} \dot{\theta} - m_{leg} g r_{leg x}$$

Donde:

$$I_{leg} = 105926'0527 \text{ kgmm}^2$$

C: toma un valor entre 0 y 1.

t_s : es el periodo que se desea para que el robot realice el movimiento.

θ : ángulo que van formando las patas a lo largo del movimiento.

θ_s : ángulo que se desea que formen las patas al final del movimiento.

$\dot{\theta}$: derivada del ángulo que van formando las patas en función del tiempo.

$$m_{leg} = 1'94576 \text{ kg}$$

g : es la aceleración de la gravedad.

$$r_{leg x} = 200 \sin(\theta - \gamma)$$

4.6.2 Pata apoyada

$$\begin{aligned}\tau_h &= k_{po} \left(\gamma - \frac{\theta_s}{2} \right) + k_{do} \dot{\gamma} - m_{robot} |\vec{r}_{robot} \times \vec{g}| - \tau_f = \\ &= I_{robot} \left(\frac{t_s}{7} \right)^{-2} \left(\gamma - \frac{\theta_s}{2} \right) + 2 \sqrt{I_{robot}^2 \left(\frac{t_s}{7} \right)^{-2}} \dot{\gamma} - m_{robot} g r_{robot\ x} - \tau_f\end{aligned}$$

$$\tau_h = I_{robot} \left(\frac{t_s}{7} \right)^{-2} \left(\gamma - \frac{\theta_s}{2} \right) + 2 I_{robot} \left(\frac{t_s}{7} \right)^{-1} \dot{\gamma} - m_{robot} g r_{robot\ x} - \tau_f$$

Los cambios respecto a la otra ecuación son:

$$I_{robot} = 316526'5604 + 1'94576((-410 \sin \gamma - 210 \sin(\theta - \gamma))^2 + (410 \cos \gamma - 210 \cos(\theta - \gamma))^2)$$

γ : es el ángulo que forma la pata apoyada con el eje vertical.

$\dot{\gamma}$: es la derivada del ángulo γ

$$m_{robot} = 4'92543kg$$

$$r_{robot\ x} = 330'99 \sin \gamma + 79'01 \sin(\theta - \gamma)$$

τ_f : es el valor del par aplicado a la pata en el aire.

CAPÍTULO 5

Modelado y simulación con Adams

5 MODELADO Y SIMULACIÓN CON ADAMS

5.1 INTRODUCCIÓN AL MANEJO DE ADAMS VIEW

El programa MSC.ADAMS VIEW es una herramienta que nos permite el modelado en tres dimensiones de mecanismos, y una vez diseñados simular diferentes situaciones, obteniendo los resultados como animaciones y los valores que se deseen medir en forma de gráfica de manera que podamos trabajar con ellos posteriormente.

Se nombran a continuación a modo de guía las características básicas del programa y en especial las empleadas en nuestra simulación [9].

Para comenzar, en la pantalla de inicio se nos da la opción de comenzar con un nuevo proyecto (al que le damos el nombre deseado), abrir un archivo ya existente, importar un archivo o salir; además elegiremos si queremos que en nuestro diseño se tenga en cuenta la fuerza de la gravedad y las unidades que queremos emplear.

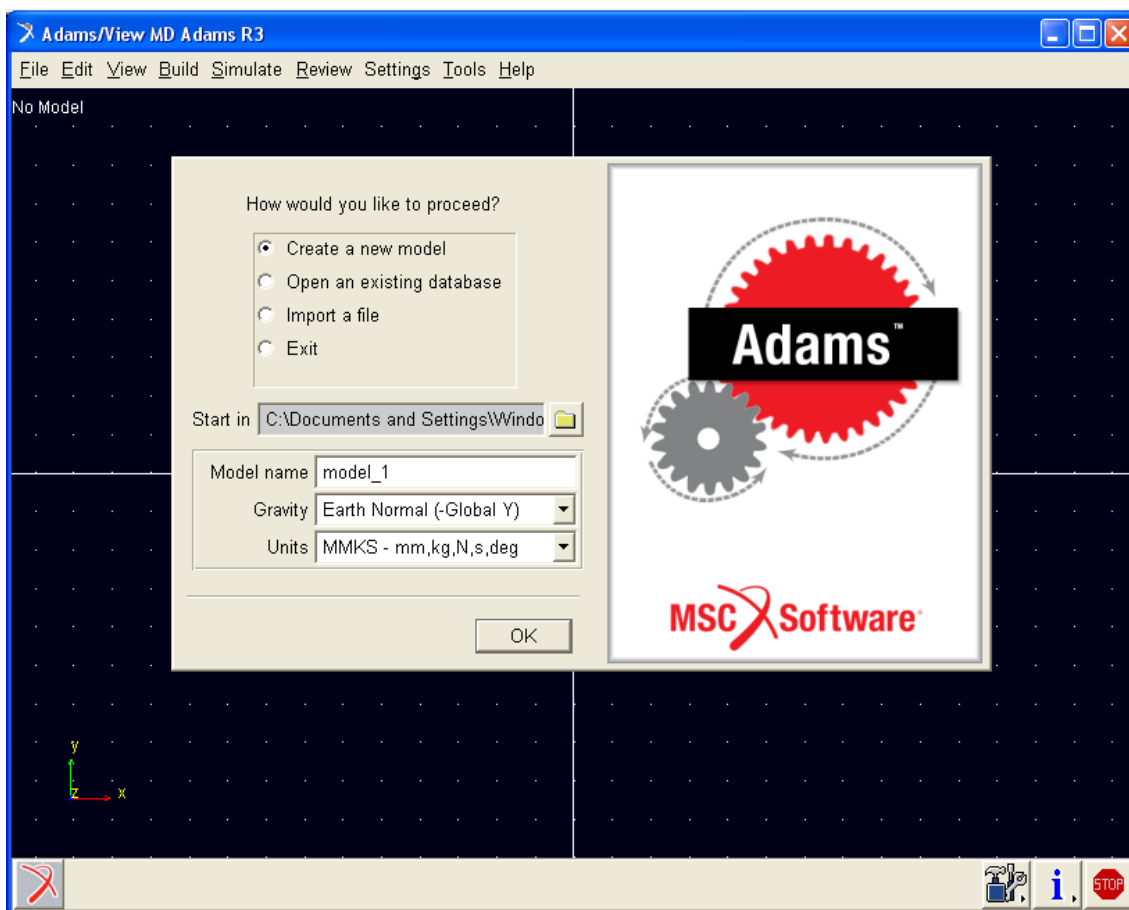


Figura 5.1 Menú inicio Adams View.

5.1.1 Botones básicos

El elemento que se emplea para realizar las diferentes operaciones será la barra denominada “Main Toolbox”.

Podemos dividir la herramienta en cuatro partes diferentes:

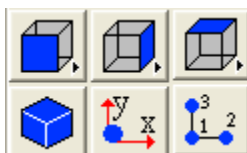
1-Parte superior: los botones que se encuentran en la parte superior serán los empleados para diseñar piezas, establecer movimientos y restricciones y generar las simulaciones, más adelante se explicará cada botón de forma detallada.



2-Botones vistas.



Con estos botones podemos movernos en la plantilla de diseño de diferentes maneras, como puede ser rotando o trasladandonos sobre algún eje, o hacer zoom sobre las piezas.



Con estos otros se podrá variar el plano de trabajo, frontal o posterior, izquierda o derecha, superior o inferior o en isométrico. Además se podrá situar el plano de trabajo en un cuerpo según sus ejes X-Y ó alinear el plano por tres puntos.

3-Botones entorno.



Con estos tres botones podemos modificar el fondo de pantalla, dividirla o dar más opciones.

4-Botones textuales.

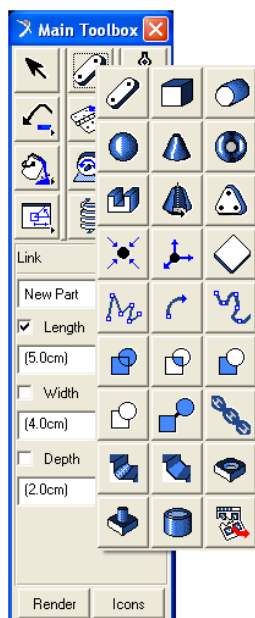
Las funciones de estos cuatro botones serán:

- Grid: activa o desactiva la malla de trabajo.
- Depth: se cambia la perspectiva de profundidad.
- Render: se renderizan los elementos.
- Icons: hace visibles o no los iconos en los elementos.



5.1.2 Diseño de piezas

Se explican a continuación los botones de la parte superior de la tabla de herramientas, siendo mayor la descripción de aquellos más importantes para nuestro diseño



Si pinchamos con el ratón en este botón nos aparece el menú desplegable de la figura 5.2.

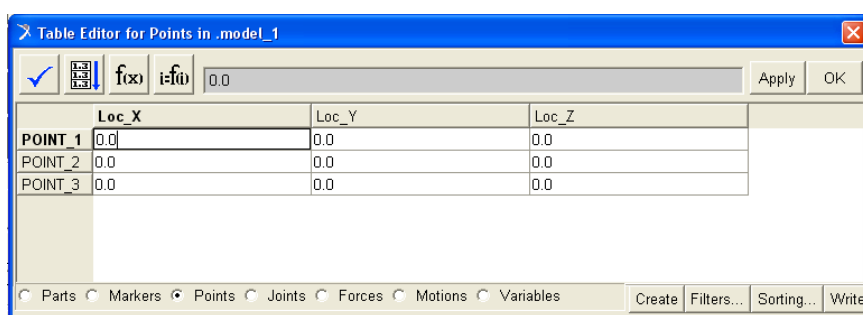
Las **tres primeras filas** corresponden al diseño de sólidos. Tenemos formas como eslabones, paralelepípedos, cilindros, esferas...Será importante definir sus dimensiones introduciéndolas directamente o a partir de puntos creados anteriormente. Para un eslabón por ejemplo determinaremos la longitud, anchura y profundidad de la pieza.

Figura 5.2 Menú creación de piezas.

En la **cuarta fila** encontramos dos botones importantes que son los de crear puntos y marcadores:



Point table (figura 5.3): nos aparece una tabla con los puntos que hemos creado y nos da la posibilidad de crear nuevos puntos o eliminar alguno de los existentes.



	Loc_X	Loc_Y	Loc_Z
POINT_1	0.0	0.0	0.0
POINT_2	0.0	0.0	0.0
POINT_3	0.0	0.0	0.0

Figura 5.3 Tabla de puntos.



Markers: son puntos de referencia que aparecen en las piezas, normalmente en su centro de masas y en sus puntos más significativos, como sus extremos. Son utilizados principalmente para dar posición a una pieza nueva respecto de otra que ya existe, para realizar medidas, para posicionar juntas, motores y para desplazar o rotar piezas utilizándolos como referencia.

Con los botones de la **sexta y séptima fila** podemos ejecutar operaciones booleanas:



Unir: Fusiona dos sólidos de manera que se transforma en uno sólo. Las dimensiones de los sólidos que lo componen no se pueden modificar.



Pegar: Une dos sólidos manteniendo las posiciones uno respecto de otro. Las dimensiones de los sólidos sí se pueden modificar.



Intersecar: Al aplicar esta función el sólido resultante es la intersección de los dos sólidos definidos.



Cortar: El sólido elegido en primer lugar es cortado por el elegido en segundo lugar, desapareciendo entonces el segundo sólido.



Con los botones de las **dos últimas filas** podemos realizar modificaciones sobre las piezas como pueden ser redondeos o taladros.



Con el **último botón** abriremos el menú de la figura 5.4, similar al explicado pero con la posibilidad de trabajar de forma independiente a la MAIN TOOLBOX.

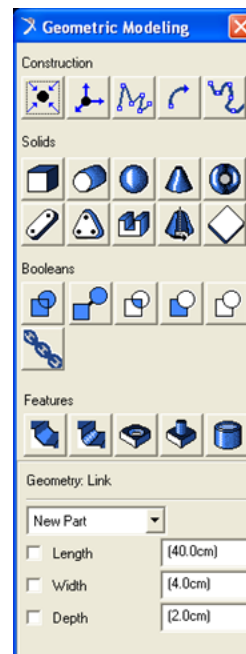
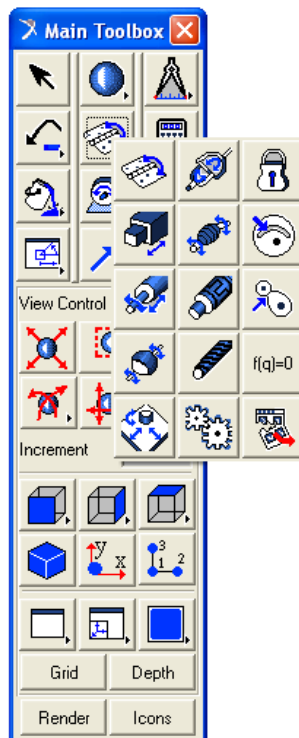


Figura 5.4 Menú independiente de creación de piezas.

5.1.3 Diseño de juntas y generadores de movimiento



Al igual que en el caso anterior si pinchamos el último botón nos aparecerá un menú independiente (figura 5.6):

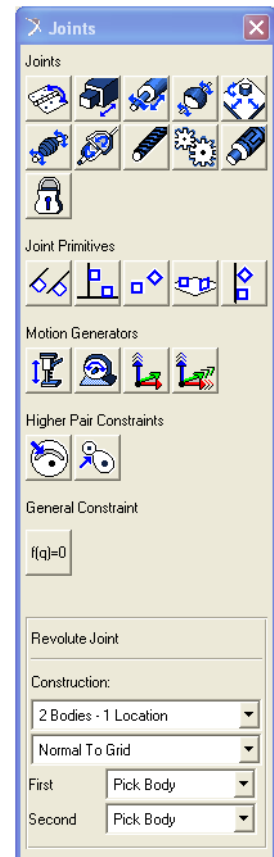


Figura 5.5 Menú de creación de juntas y generadores de movimiento.

Figura 5.6 Menú independiente de creación de juntas y generadores de movimiento.

Juntas (Joints): seleccionaremos unas o otras en función de los grados de libertad que queramos restringir. Las más utilizadas en nuestro proyecto serán las juntas de rotación y fijas



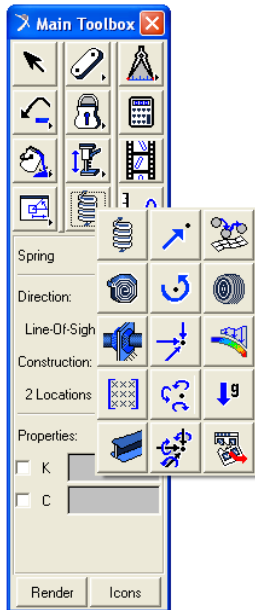
Junta de rotación: permite únicamente el giro en una dirección, restringiendo el resto de movimientos.



Junta fija: restringe todos los movimientos, es decir, los grados de libertad de un sólido respecto a otro serán cero.

Actuadores (Motion Generators): generan movimiento en el sólido, tenemos la opción de incluir motores lineales, rotativos o actuadores en la dirección que deseemos.

5.1.4 Aplicación de fuerzas y contactos



Si pinchamos en el último botón nos aparecerá el menú independiente de la figura 5.8

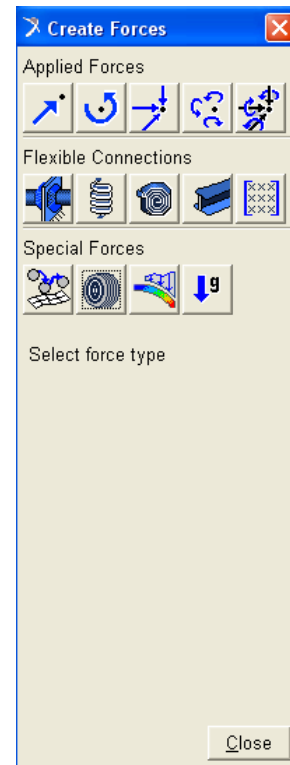


Figura 5.7 Menú de creación de fuerzas y contactos.

Figura 5.8 Menú independiente de creación de fuerzas y contactos.



Contacto: es una herramienta importante con la que le indicamos al programa entre que superficies deseamos que exista contacto o en el caso contrario, no establecer contacto y evitar así choques entre piezas que no queremos tener en consideración. Nos aparecerá el siguiente menú en el que indicaremos los sólidos a contactar y la fuerza de rozamiento que habrá entre ellos.

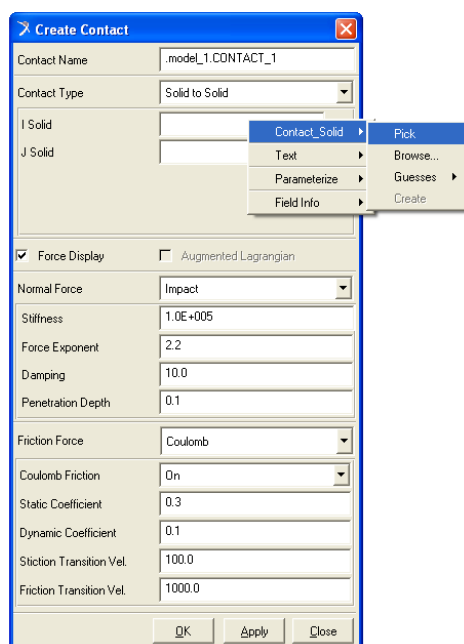


Figura 5.9 Creación de contacto.

Fuerzas: podemos generar fuerzas y pares puntuales sobre el sólido, incluir o no la fuerza de la gravedad y aplicar fuerzas y pares amortiguados; pero lo más importante en este caso es la posibilidad de generar par entre dos sólidos.

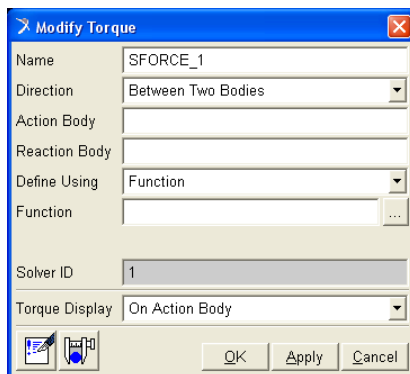
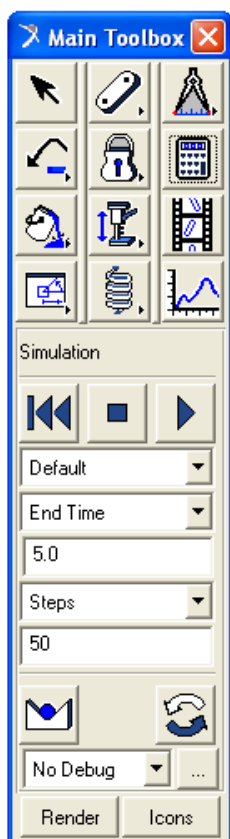



Figura 5.10 Creación de par.


El valor del par no tiene necesariamente que ser un número, el programa nos permite incluir funciones y hacer referencia a medidas de nuestro objeto. Podemos por ejemplo hacer funciones senosoidales, hacer que el par aumente a medida que aumenta un determinado ángulo o incluso funciones temporales en las que el par aparezca únicamente en un determinado periodo.

5.1.5 Simulación



Una vez finalizado el diseño podemos comprobar cómo se comportan los objetos con las solicitaciones impuestas. Para iniciar la simulación pincharemos en  y la Main Toolbox pasará a ser como se muestra en la figura 5.11:

En los valores de End Time y Steps hay que especificar cuánto queremos que dure nuestra simulación y cuantos pasos queremos que incluya. A mayor número de pasos más precisión.

Con los tres botones superiores podemos detener o reanudar la simulación y con este  repetimos la última simulación efectuada.

Si queremos comprobar el equilibrio de nuestro diseño pulsaremos:





Podemos visualizar la animación de una manera más detallada y con diferentes opciones de reproducción empleando:



Figura 5.11 Menú de simulación.

5.1.6 Toma de datos

Una vez hecha la simulación podemos efectuar diferentes medidas, tenemos dos opciones.

- Si utilizamos los botones   se pueden medir ángulos, desplazamientos, velocidades y aceleraciones.
- Si lo que buscamos son características de un punto concreto, pincharemos sobre él con el botón derecho seleccionando “Measure” y en el menú que aparece seleccionaremos la magnitud a medir y el eje en el que lo queremos.

5.1.7 Postprocesador



El postprocesador es una herramienta que permite un estudio detallado de los resultados que se han obtenido mediante las simulaciones. Además permite visualizar animaciones para su posterior exportación.

Se puede acceder a él mediante el acceso directo o directamente desde ADAMS View de una manera más cómoda siguiendo los pasos siguientes:

Cuando se toma una medida de cualquier tipo se pulsará sobre ella con el botón derecho y se seleccionará “Transfer To Full Plot” de la siguiente forma:

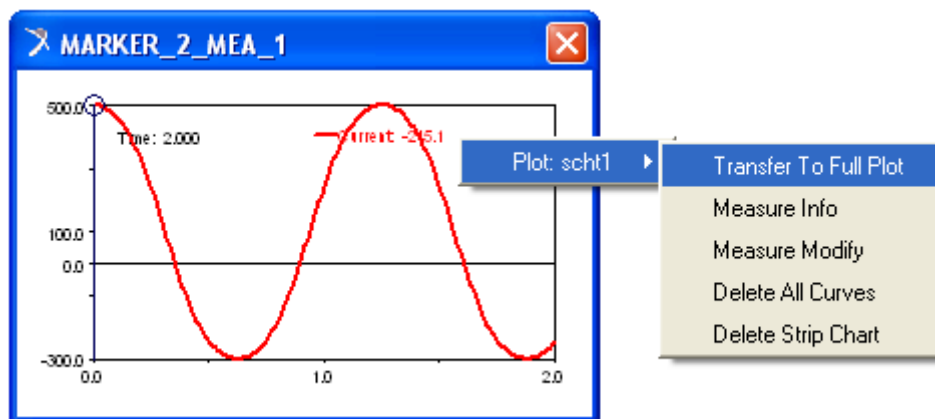


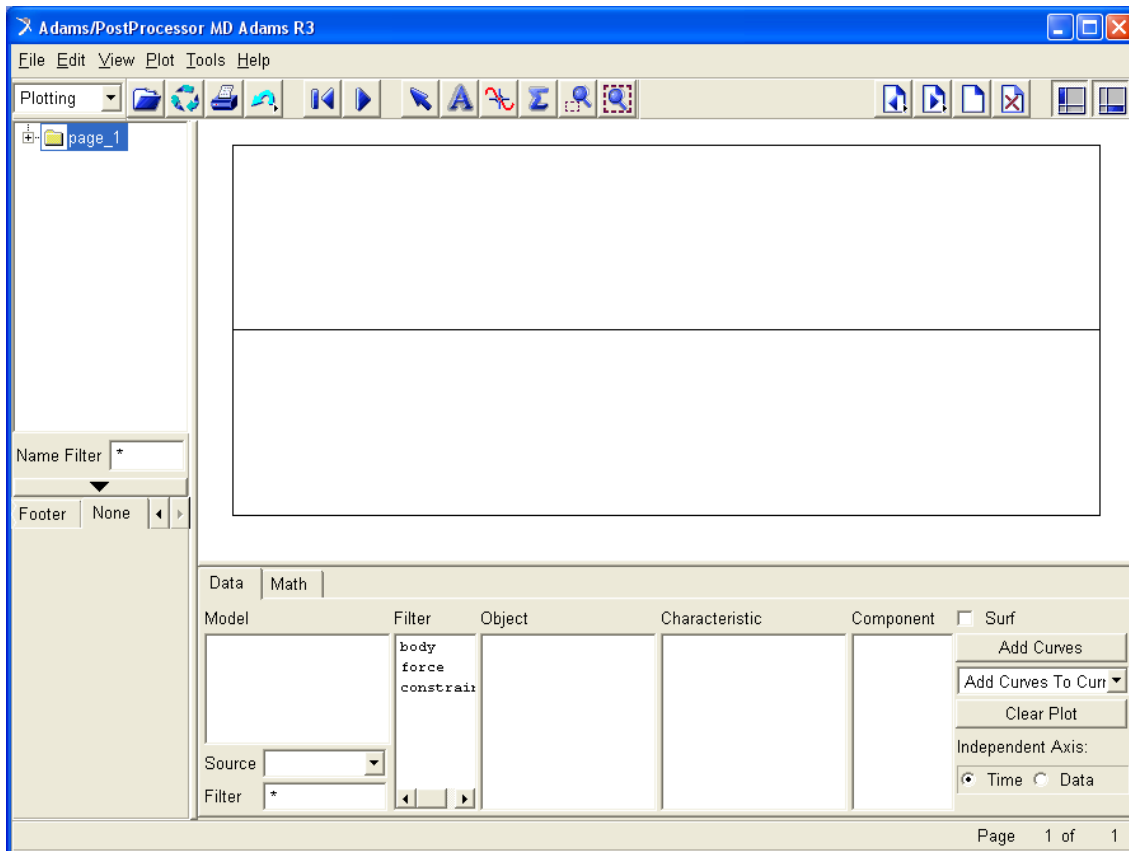
Figura 5.12 Paso de medida del postprocesador.

Una vez hecho se abrirá el postprocesador automáticamente. Todo lo que se guarde de esta forma a partir de ahora quedará guardado en la misma base de datos que estemos utilizando de manera que siempre que esté la base de datos abierta en la que la cual se esté trabajando se podrá acceder directamente al postprocesador pinchando en el siguiente icono de la barra de tareas:

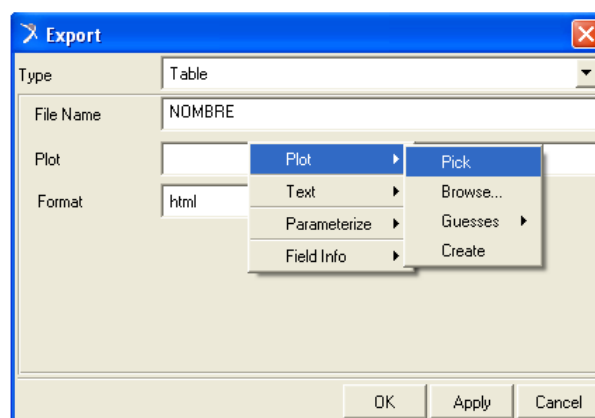


ADAMS View con:



**Figura 5.13 Postprocesador**

Para exportar datos se seleccionará en “File” la opción “Export”, “Table”. Entonces aparecerá el cuadro de la figura 5.14, donde se definirá el nombre del archivo y la serie de datos que se desea exportar. Para ello, tal y como se muestra, a continuación se pincha directamente sobre el gráfico.

**Figura 5.14 Exportación de datos y archivos**

Es importante antes de aceptar, haber elegido una carpeta de destino, que será en principio la que se definió al iniciar el nuevo modelo. Si no se conoce se elegirá en “File”, “Select Directory”.

Para cargar animaciones, se pulsará con el botón derecho en cualquier parte de la gráfica y se seleccionará “Load Animation”, cargándose automáticamente la animación.

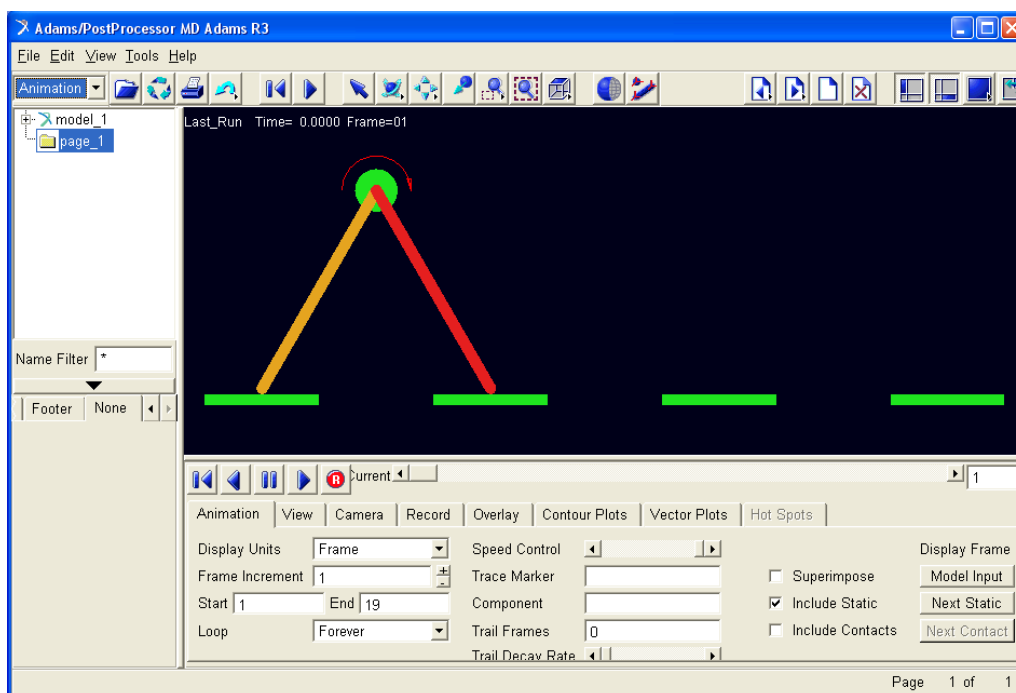


Figura 5.15 Carga y visualización de animaciones en el Postprocesador.

Ahora se podrá reproducir, con el cuadro que se muestra, cambiar la cámara, la vista, la calidad y grabar vídeos para su exportación, lo que se hará de la siguiente forma:

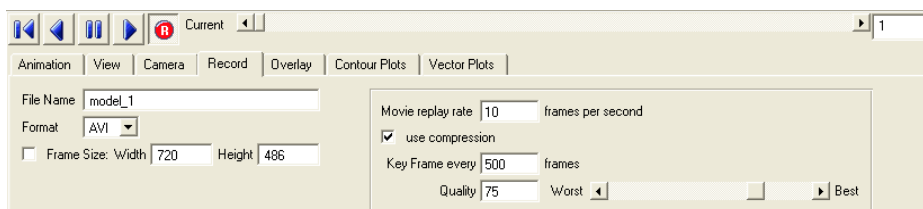


Figura 5.16 Grabación de animaciones.

Primero, en “File Name” se seleccionará el nombre de la animación y se definirá la calidad de la misma.

Se pulsará el icono de  grabar y posteriormente, el play. 

El archivo se grabará en la carpeta seleccionada por defecto en formato AVI.

5.2 MODELADO DEL ROBOT

Se detallan a continuación los diferentes pasos seguidos en el proceso de creación del modelo.

5.2.1 Cambio en el diseño

El primer paso será modelar las piezas, pero antes de comenzar surge un problema, ya que el modelo bípedo funciona correctamente de manera teórica y en simulaciones bidimensionales, pero en una simulación real la estabilidad se ve comprometida al sostenerse el robot en un punto de la pata apoyada (ZMP). Para solucionar este problema se puede optar por varias alternativas, como sería la inclusión de un pie o elemento similar en la parte inferior de la pata, o la que parece más adecuada por su total corrección de este problema, que sería la transformación de lo que hasta ahora era un modelo bípedo en un modelo cuadrúpedo, con simetría en la forma.

El diseño será similar al explicado en la parte teórica pero dividiendo la anchura de las patas en dos, de manera que no se modifican ni las masas ni las inercias ya calculadas a la hora de introducirlas en las ecuaciones de control.

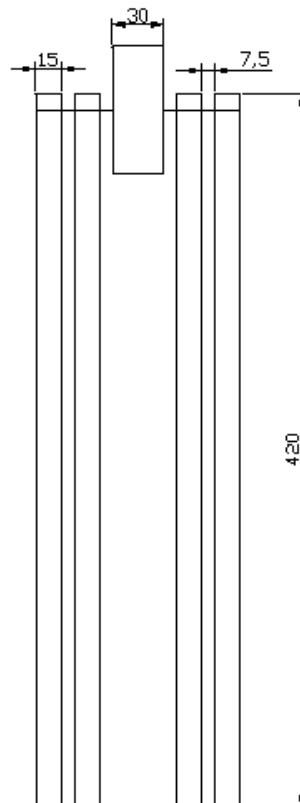


Figura 5.17 Nuevo diseño del robot.

5.2.2 Modelado

5.2.2.1 Creación de puntos

Para facilitar el diseño de las piezas se crean puntos que nos sirven de guía:

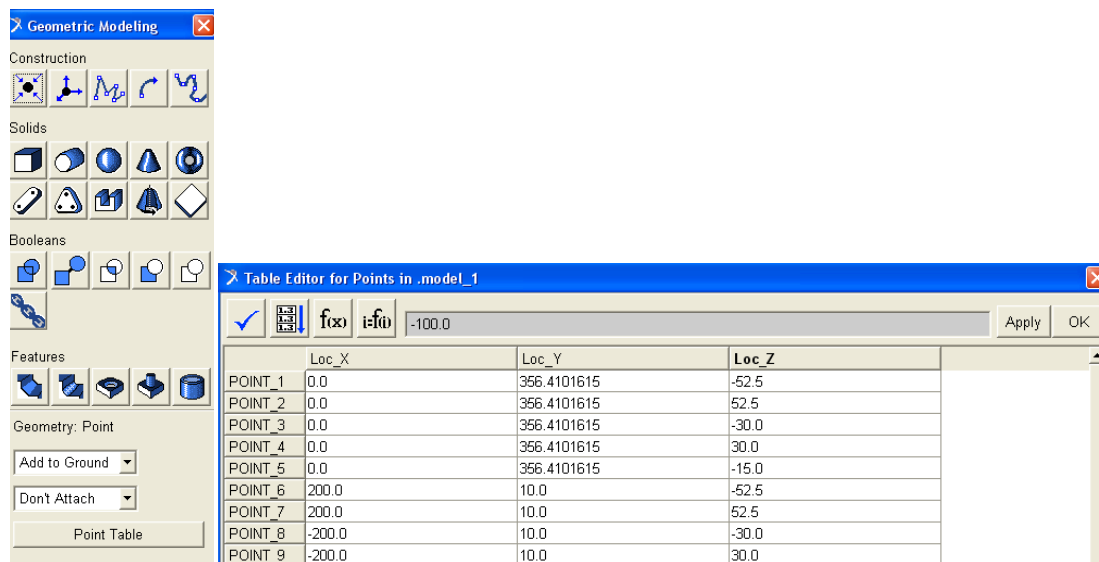


Figura 5.18 Tabla de puntos.

La posición inicial elegida para que el robot comience el movimiento es con un ángulo entre patas de 60° , y 30° entre la pata apoyada y la vertical. Por lo tanto si la longitud entre los puntos que forman la pata es de 400mm , la altura de estos será $y = 400 \cos 30 = 346.4101615\text{mm}$, añadiéndose otros 10mm debido a la parte redondeada de radio 10mm que forma la parte inferior de la pata y que entra en contacto con el suelo, quedando por lo tanto las componentes y de los puntos superiores como 356.4101615mm y las de los puntos inferiores 10mm . En el caso de la componente horizontal tenemos $x = 400 \sin 30 = 200\text{mm}$, siendo positivos en el caso de las patas apoyadas que están más adelantadas, y negativo en las traseras.

En cuanto a la componente z los puntos se elaboran en función de la profundidad de los elementos y dejando una separación entre estos de 7.5mm .

Tenemos por lo tanto los puntos 1 y 2 que corresponden a los puntos superiores de las patas apoyadas, los 3 y 4 correspondientes a los puntos superiores de las patas traseras. El punto 5 se define para el disco. Los puntos 6 y 7 los inferiores de las patas apoyadas y por último el 8 y 9 para la parte inferior.

Se pueden añadir más puntos en el momento que se desee por lo que posteriormente se añadirán los que forman las plataformas sobre las que se moverá el robot.

5.2.2.2 Creación de piezas

Una vez definidos los puntos, la creación de las piezas es sencilla. Se eligen elementos del tipo LINK para las patas marcando en el menú las propiedades que queremos definir y uniendo los puntos que la componen.

Se escoge una longitud de 40 cm, anchura de 2 cm y profundidad de 1'5 cm (figura 5.19), y se unen los puntos 1-6, 2-7, 3-8 y 4-9.



En el caso del disco el procedimiento es similar, fijamos las propiedades (figura 5.20) y nos situamos en el punto 5 pero en lugar de unirlo con otro punto lo dirigimos hacia el sentido z positivo, quedando con esta longitud simétrico respecto a todos los ejes.

Se incluyen dos cilindros que unen cada pareja de patas a modo de ejes pero de radio despreciable de manera que sirven de unión entre piezas pero no influyen en las propiedades del conjunto total.

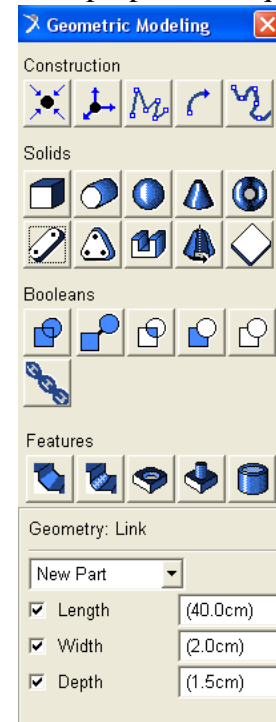


Figura 5.19 Creación de patas.

Figura 5.20 Creación del disco.

El conjunto de piezas creadas vistas desde un plano tridimensional queda del siguiente modo:

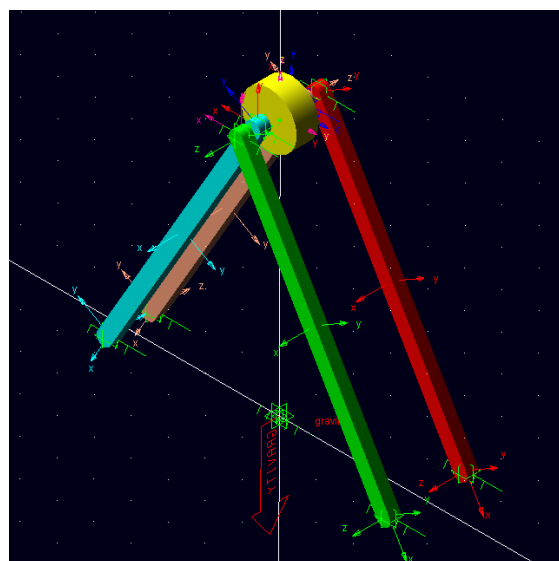


Figura 5.21 Vista tridimensional del conjunto.

5.2.2.3 Uniones entre piezas

Para unir las diferentes piezas creadas podemos proceder de varias maneras: la primera sería uniendo las piezas de forma definitiva, de manera que cada pareja de patas y su eje pasan a formar una única pieza; y la segunda y que en este caso nos resulta más práctica para realizar medidas y si fuera necesario ciertos cambios es establecer relaciones de empotramiento entre esas piezas.

Entre los dos ejes y a su vez estos con el disco se permite únicamente el giro, impidiéndose el resto de movimientos.

Cambiando además el color de las piezas resulta más intuitivo reconocer cada elemento del conjunto.

Al finalizar este paso tenemos:

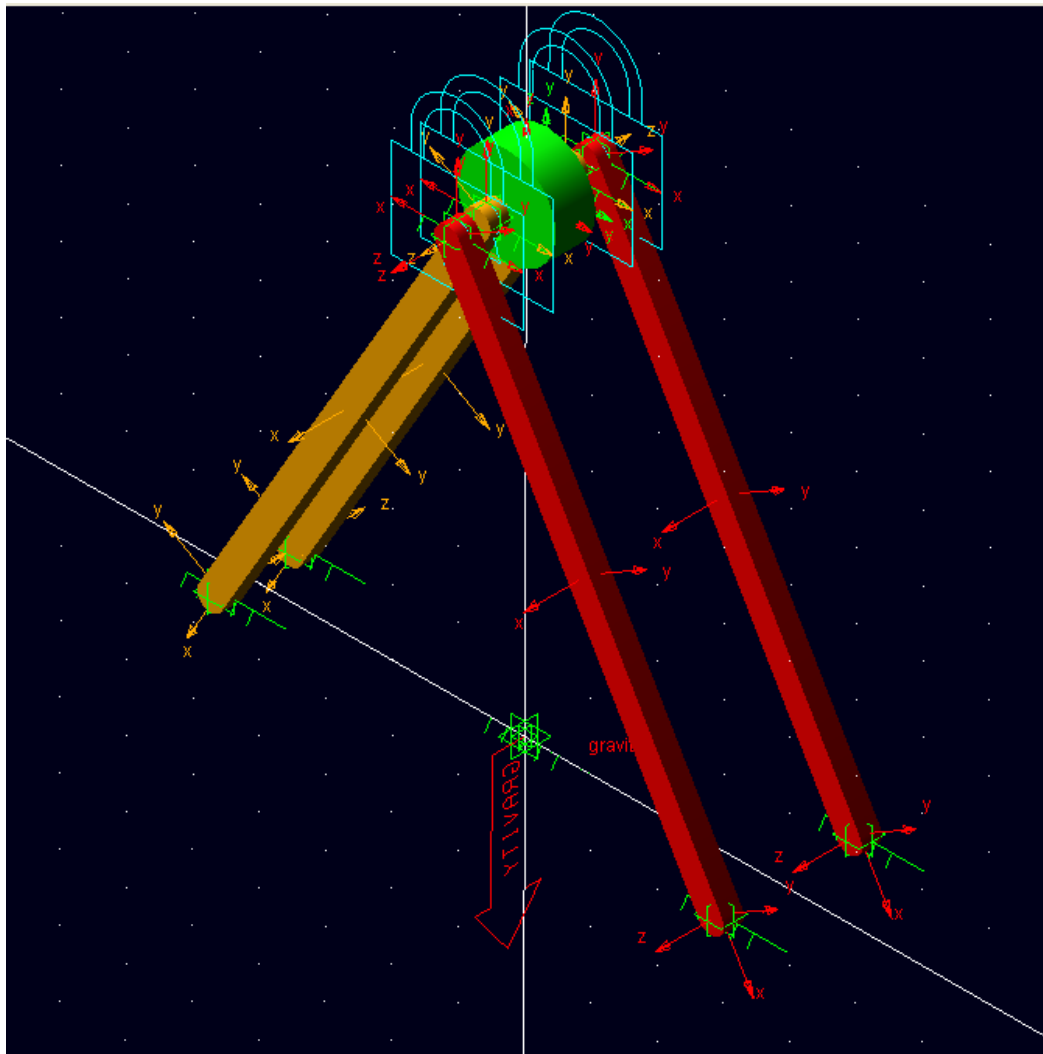


Figura 5.22 Uniones entre piezas.

5.2.2.4 Plataformas y contactos

Una vez se ha modelado el conjunto robot, se procede a diseñar la superficie sobre la que ha de caminar.

El primer paso es crear los puntos que sirven de guía

POINT_10	100.0	-20.0	-100.0
POINT_11	500.0	-20.0	-100.0
POINT_12	-300.0	-20.0	-100.0
POINT_13	900.0	-20.0	-100.0

Figura 5.23 Puntos guía para las plataformas.

Los valores de x son los correspondientes para crear cuatro plataformas, dos para que apoyen las patas al comienzo del movimiento y otras dos para que se apoyen en los sucesivos pasos.

Se escogen plataformas con forma prismática de lados 20 cm y 2 cm de profundidad, de ahí los valores de y y z en la tabla de puntos. Se establecen relaciones de empotramiento entre cada una de las plataformas y el suelo (ground) como se observa en la figura 5.24.

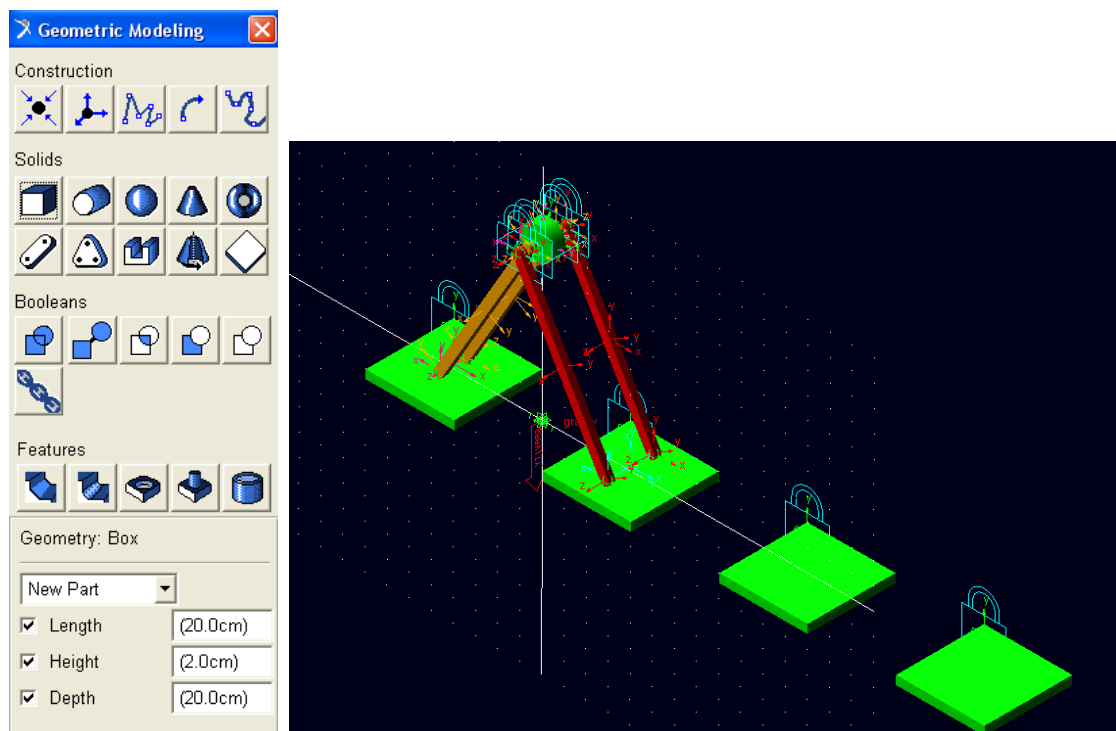


Figura 5.24 Creación y vista tridimensional de las plataformas.

El siguiente paso es crear los contactos existentes entre las patas y las plataformas. Solo se relaciona cada pata con las plataformas en la que se apoya al comienzo del movimiento o se apoyará a lo largo de este, pero no con el resto de plataformas para evitar colisiones que modifiquen las trayectorias de las patas.

Se definen por la tanto ocho contactos: el 1 y 2, 7 y 8 entre las patas rojas con las plataformas 2 y 4, mientras que los 3 y 4, 5 y 6, corresponden a las patas marrones con las plataformas 3 y 1 respectivamente (figura 5.26) .

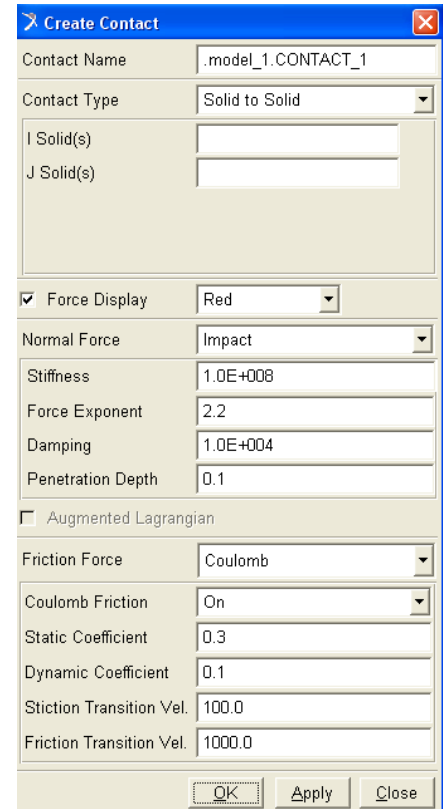


Figura 5.25 Creación de contactos.

Los coeficientes de rozamiento se modificarán posteriormente en función de las necesidades de la simulación

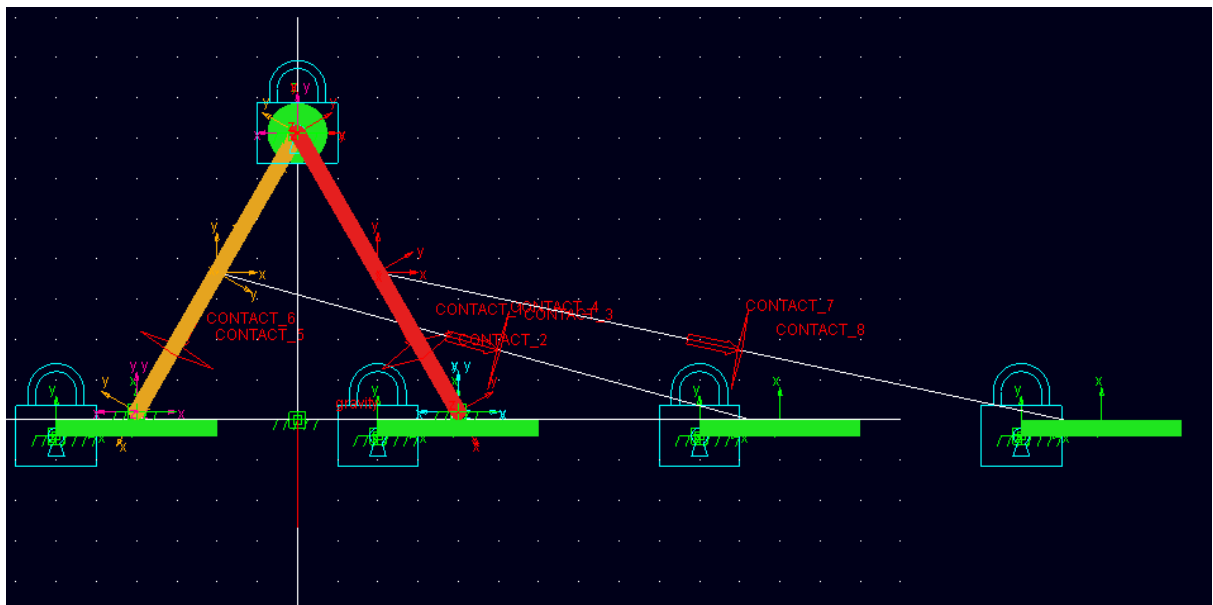


Figura 5.26 Contactos entre piezas y plataformas.

5.2.3 Introducción de pares

Una vez resueltos los aspectos de modelado de las piezas se procede a la inclusión de los pares que controlarán el movimiento y a observar la exactitud con la que estos funcionan.

Este punto del proceso consiste en introducir las ecuaciones de control, que se presentaron en el análisis teórico, en el modelo físico ya creado. La forma final de estas ecuaciones resultaba:

Pata en el aire

$$\tau_f = I_{leg} \left(\frac{c t_s}{7} \right)^{-2} (\theta - \theta_s) + 2I_{leg} \left(\frac{c t_s}{7} \right)^{-1} \dot{\theta} - m_{leg} g r_{leg x}$$

Pata apoyada

$$\tau_h = I_{robot} \left(\frac{t_s}{7} \right)^{-2} \left(\gamma - \frac{\theta_s}{2} \right) + 2I_{robot} \left(\frac{t_s}{7} \right)^{-1} \dot{\gamma} - m_{robot} g r_{robot x} - \tau_f$$

Nota: antes de comenzar con la implementación de las ecuaciones del par es importante conocer las unidades con las que se está trabajando. Si realizamos un análisis dimensional observamos que nuestras unidades para el par son $kg * mm^2 * s^{-2}$ o lo que es lo mismo $mN * mm$. Será necesario modificar por lo tanto las unidades predefinidas en el programa, para ello se realiza la siguiente operación:

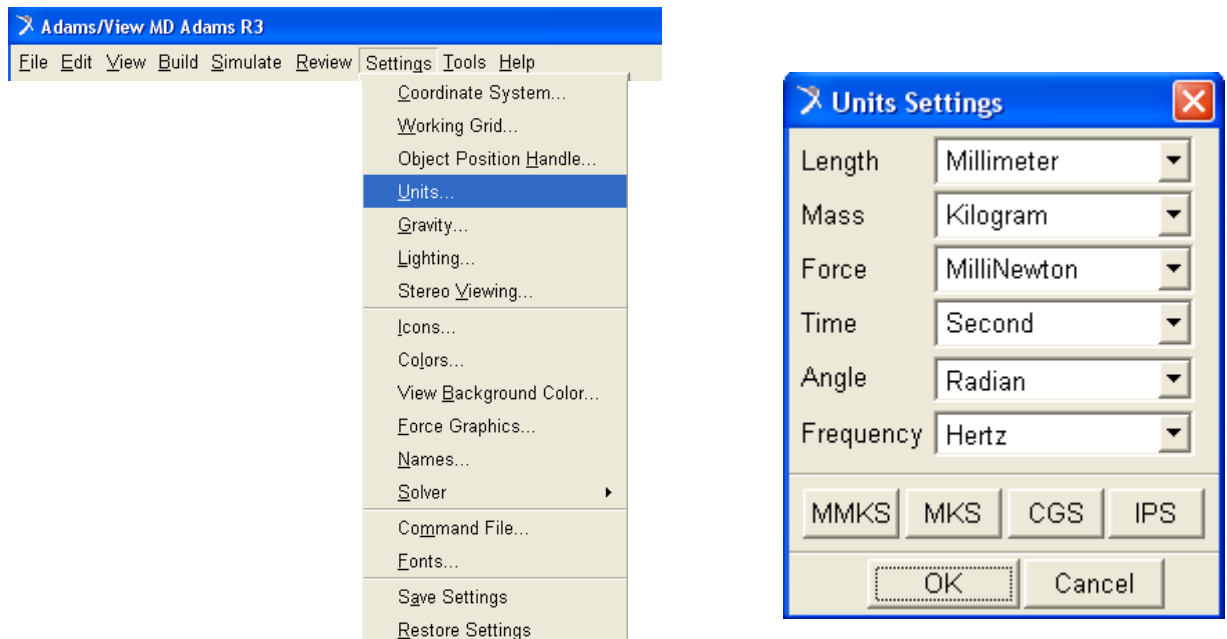


Figura 5.27 Menú unidades.

Únicamente es necesario cambiar las unidades de fuerza de Newton a MiliNewton y las de ángulo de Degree a Radian.

Realizado ya el cambio de unidades necesario se pueden introducir los dos pares que actúan.

En el menú de fuerzas se escoge Torque y dentro de sus opciones la que nos permite establecer pares entre dos cuerpos, ya que son los utilizados en nuestro modelo (figura 5.28). Se debe señalar cuál es el objeto de acción, cuál el de reacción y ambos puntos de acción reacción.

Para el par que llamaremos Tf, el que actúa sobre la pata en el aire, se elige como elemento actuante el eje de las patas libres y como elemento de reacción las patas apoyadas. Como puntos de acción y reacción se toman ambos centros de masas de las piezas.

En el caso del par que controla la pata apoyada, Th, el elemento que actúa es el eje de las patas apoyadas y el elemento que lo recibe es el disco situado en la parte central. Al igual que en el caso anterior los puntos de acción-reacción son los centros de masas de estos elementos.

Ambos pares se introducen según la figura 5.29

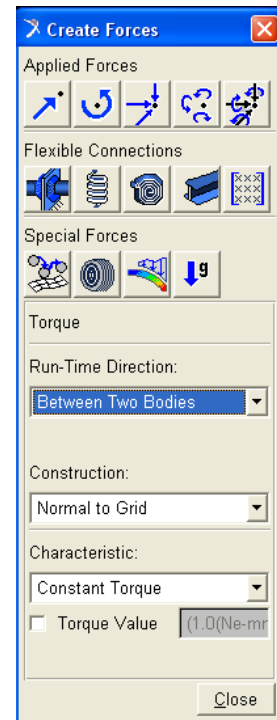


Figura 5.28 Menú creación de pares.

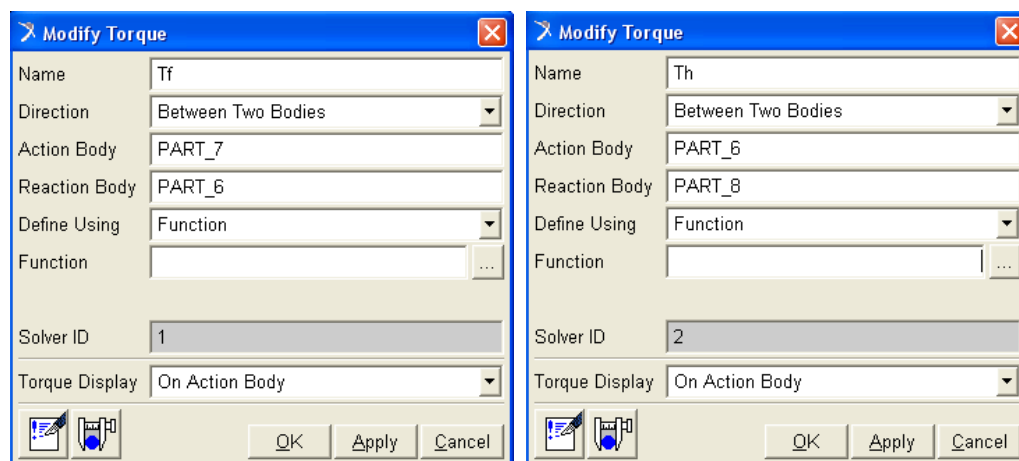


Figura 5.29 Características de los pares Tf y Th.

Faltaría únicamente incluir las funciones de control para estos pares, pero antes hay que conocer con detalle cada elemento que lo forma y saber cómo introducirlo en el programa.

Una ventaja del programa es que nos permite introducir dentro de la función resultados de distintas mediciones durante la simulación, o crear una subfunción que nos aporte resultados necesarios para nuestra ecuación.

La manera de crear una medida o subfunción es la siguiente:

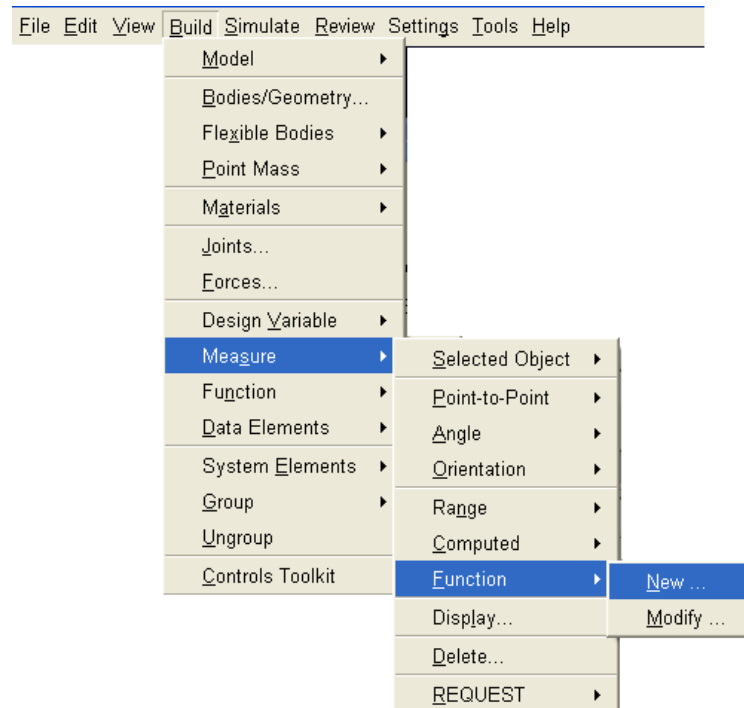


Figura 5.30 Ruta para crear una función.

Introduciendo en el siguiente cuadro la ecuación que se necesite y el nombre que se desee dar.

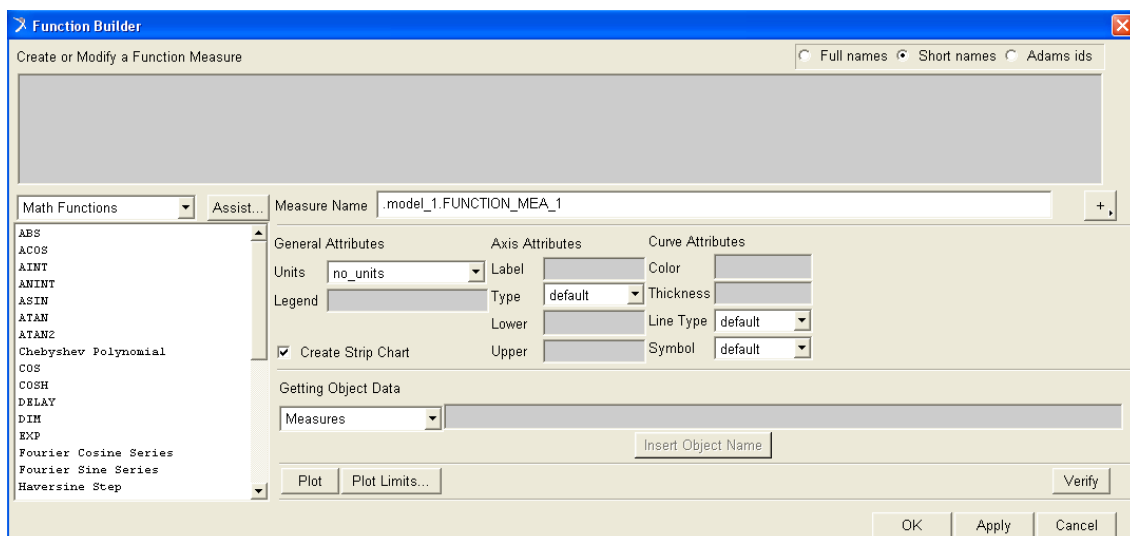


Figura 5.31 Menú creación de función.

En el caso del creador de funciones del par el menú es similar.

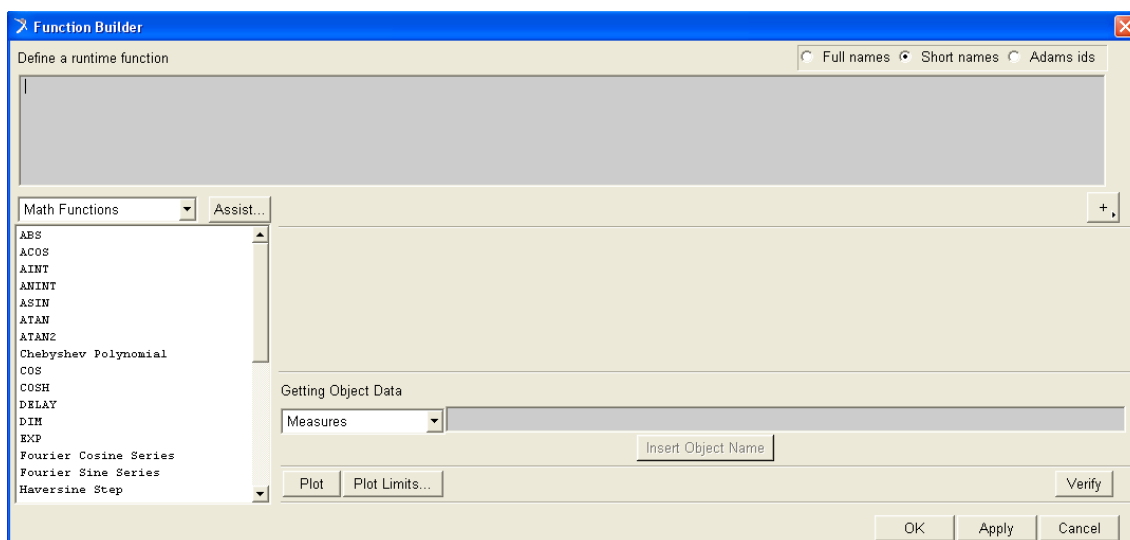


Figura 5.32 Menú creación de par.

Si se quieren utilizar los medidores definidos en la otra parte seleccionaremos en “Getting Object Data” la opción de “Measures” y realizando doble click en la barra gris nos aparecerán todas las medidas que hemos realizado al modelo.

Además de seleccionar las medidas creadas como funciones podemos seleccionar también las medidas de ángulos y distancias que se obtienen con las herramientas del propio programa:

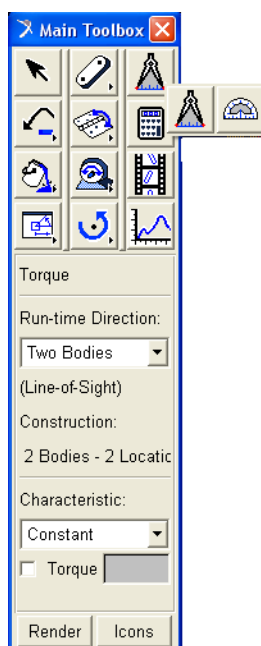


Figura 5.33 Menú de mediciones básicas.

O incluso realizar mediciones de propiedades y movimientos de cada objeto. Si pinchamos en cualquiera de las piezas con el botón derecho y seleccionamos “measure” nos aparece el siguiente menú:

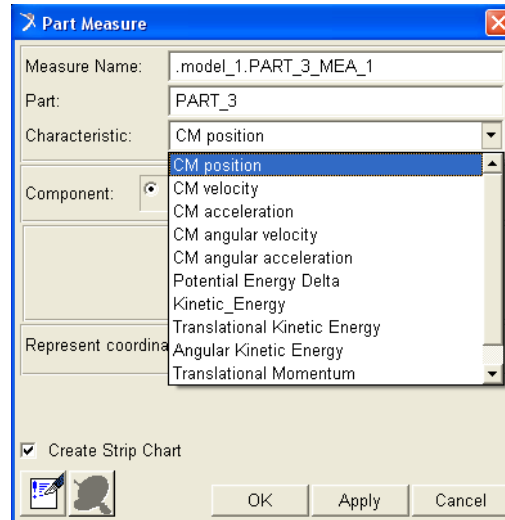


Figura 5.34 Menú avanzado de mediciones.

Como se puede ver nos permite obtener datos de su movimiento y también escoger la componente en la que se deseen estas mediciones. En nuestro caso se utilizará principalmente para medir las velocidades angulares de las patas a lo largo de la simulación.

Conocidas ya las herramientas que el programa nos ofrece se procede a introducir las ecuaciones, definiendo cada término del siguiente modo:

5.2.3.1 Pata en el aire

$$\tau_f = I_{leg} \left(\frac{c}{7} \right)^{-2} (\theta - \theta_s) + 2I_{leg} \left(\frac{c}{7} \right)^{-1} \dot{\theta} - m_{leg} g r_{leg}^2 x$$

m_{leg} : se obtiene este valor en primer lugar ya que puede ser necesario en el resto de componentes. Si se accede al menú de propiedades de cada pata tenemos que la masa de una única pata son 0.97228 kg, que al trabajar con la pareja nos queda:

$$m_{leg} = 2 \cdot 0.97228 = 1.94456 \text{ kg}$$

I_{leg} : aunque su valor es constante, la definimos como una función de medida para que sea más sencillo trabajar con las ecuaciones. El valor de la inercia respecto al centro de gravedad se puede obtener directamente del programa, y tiene un valor de 28125.351851 kgmm², pero el que se utiliza en este caso es respecto del eje sobre el que gira, por lo que aplicando Steiner nos queda:

$$I_{leg} = 28125.351851 + 1.94456 \cdot 200^2 = 105907.75 \text{ kgmm}^2$$

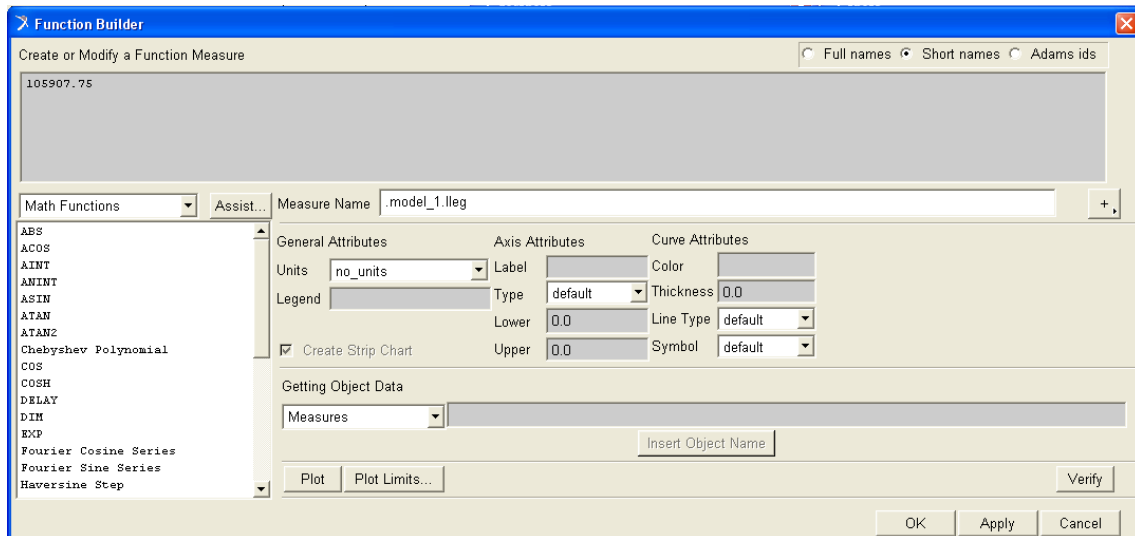


Figura 5.35 Creación de la función I_{leg} .

c: Se trata de una constante que designa el momento en el que la pata alcanza su posición final respecto al tiempo total. Se introduce directamente en la ecuación del par para simplificar el proceso, con un valor:

$$c = 1$$

ts: se crea una función de manera que sea fácil cambiar su valor en el momento de realizar las diferentes simulaciones.

θ : es el ángulo que forman unas patas con las otras, por lo que se utiliza el medidor de ángulos que proporciona el programa. Se define con tres puntos, por lo que se crean dos puntos en el punto central de cada pareja de patas, y un tercero que es el centro de masas de cualquiera de los ejes, ya que ambos coinciden.

$$\theta = theta$$

θ_s : es una de las propiedades modificables en las diferentes simulaciones por lo que para que su uso sea sencillo se crea como una función independiente, pudiendo cambiar así sus valores sin tener que modificar la ecuación del par.

$\dot{\theta}$: podemos definir esta velocidad angular como la resta de las velocidades angulares de cada pata. Ambas se obtendrían de la manera que se explicó anteriormente, introduciendo a su vez estos valores en una función para obtener el valor directo deseado en la ecuación del par. Se nombra como dertheta.

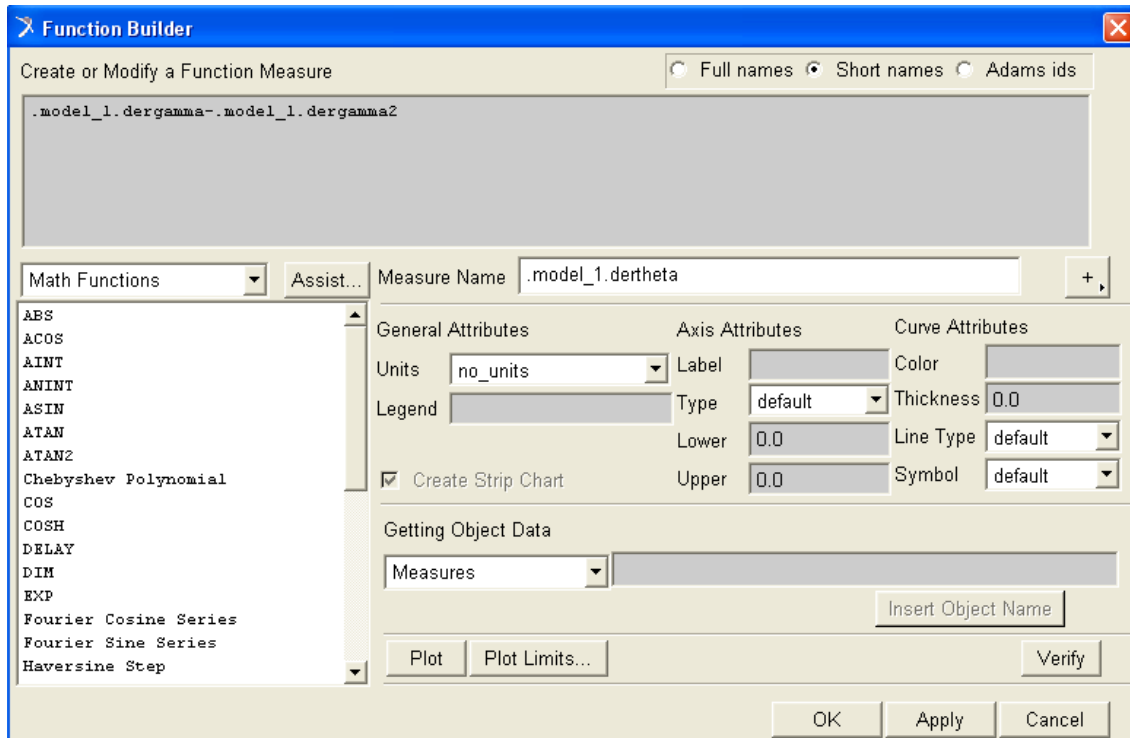


Figura 5.36 Creación de la función $\dot{\theta}$.

Dergamma se refiere a la velocidad angular de las patas apoyadas y dergamma2 a la velocidad angular de las patas en el aire.

$$\dot{\theta} = dertheta$$

$m_{leg} g r_{leg x}$: este producto no sufre modificación alguna en las diferentes simulaciones, son valores previamente calculados en el estudio teórico, por lo que podemos introducir el resultado de la multiplicación directamente en la ecuación del par. Es importante tener en cuenta que con las unidades que estamos trabajando el valor de la gravedad es de $9806 \frac{mm}{s^2}$

$$m_{leg} g r_{leg x} = 1.94456 \cdot 9806 (-200 \sin(\theta - \gamma)) = 3813671'072 \sin(\theta - \gamma)$$

Es necesario una vez definidos los diferentes términos que conforman la ecuación, establecer los criterios de signos deseados y cuáles son los establecidos por el programa, de manera que modificando los signos que sean necesarios en la ecuación de control, ambos coincidan.

El criterio de signos para los ángulos que se sigue en nuestro caso (figura 5.37), no coincide en algunos puntos con el del programa, por lo que se modifica la ecuación de control para obtener resultados correctos en las simulaciones.

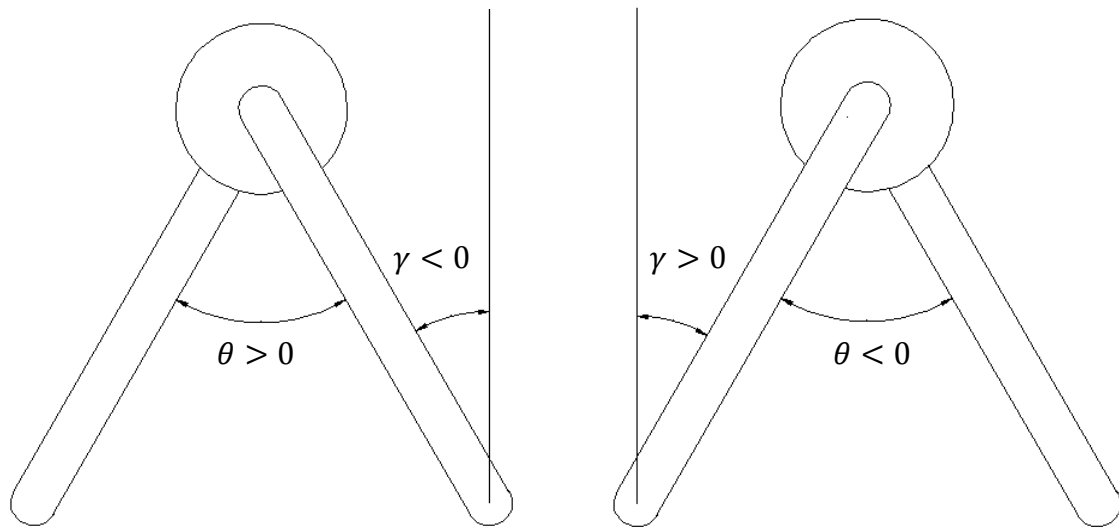


Figura 5.37 Criterio de signos escogido.

Se puede escribir por lo tanto la ecuación de control para la pata en el aire según los elementos descritos y los signos aplicados de la siguiente manera:

$$Tf = .model_1.Ileg * ((7/.model_1.ts)**2) * (.model_1.theta + .model_1.thetas) + .model_1.Ileg * (7/.model_1.ts) * .model_1.dertheta - 3813671.072 * (.model_1.theta - .model_1.gamma)$$

Aplicado en Adams como aparece en la figura 5.38.

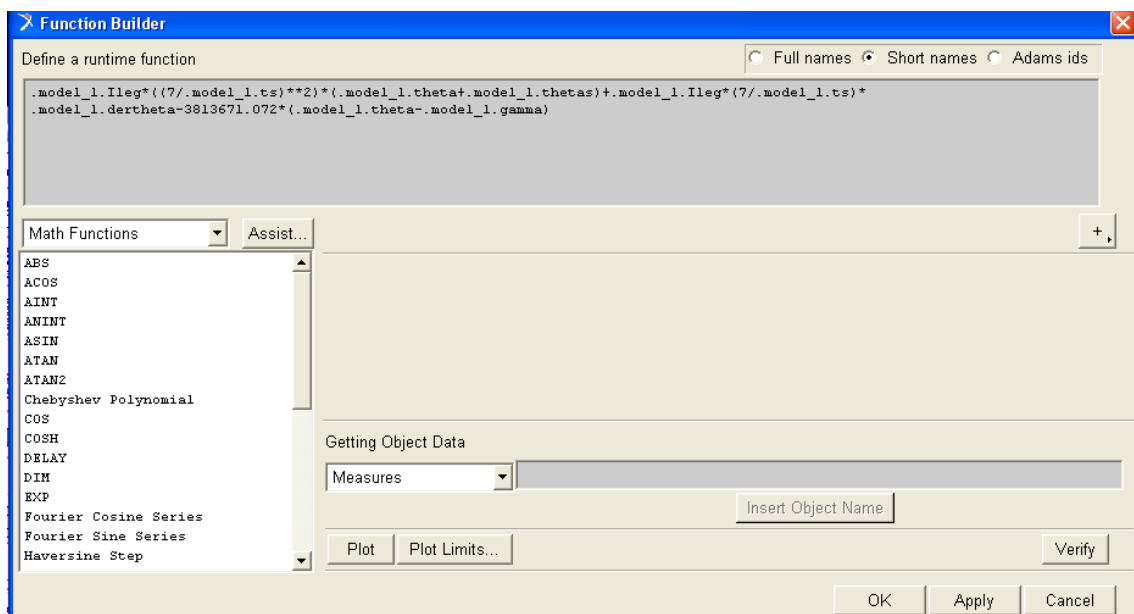


Figura 5.38 Creación de la función Tf.

5.2.3.2 Pata apoyada

$$\tau_h = I_{robot} \left(\frac{t_s}{7} \right)^{-2} \left(\gamma - \frac{\theta_s}{2} \right) + 2I_{robot} \left(\frac{t_s}{7} \right)^{-1} \dot{\gamma} - m_{robot} g r_{robot\ x} - \tau_f$$

En este caso la ecuación sigue prácticamente el modelo anterior, con algunos cambios que se explican a continuación.

I_{robot} : en este caso no se trata de una constante, ya que el valor cambia en función del ángulo que forman las patas. Hay que tener en cuenta que a diferencia de la inercia de la pata, esta se referencia respecto del ZMP.

Se crea una variable en el programa, pero antes realizamos manualmente los cálculos de su valor utilizando la información de los componentes que nos proporciona esta herramienta.

$$m_{disco} = 1'0339 \text{ kg} \quad I'_{disco} = 441'028 \text{ kgmm}^2$$

$$I_{disco} = 441'028 + 1'0339 \cdot 410^2 = 174239'618 \text{ kgmm}^2$$

$$I_{leg\ apoyada} = 28125'351851 + 1'94456 \cdot 210^2 = 113880'4479 \text{ kgmm}^2$$

$$I_{leg\ movimiento} = 28125'351851 + 1'94456((-410 \sin \gamma - 210 \sin(\theta - \gamma))^2 + (410 \cos \gamma - 210 \cos(\theta - \gamma))^2)$$

La distancia al centro de gravedad de la pata en movimiento respecto del ZMP está definida en función de las componentes x e y del estudio cinemático.

$$I_{robot} = I_{disco} + I_{leg\ apoyada} + I_{leg\ movimiento} = 316245'4178 + 1'94456((-410 \sin \gamma - 210 \sin(\theta - \gamma))^2 + (410 \cos \gamma - 210 \cos(\theta - \gamma))^2)$$

ts: es la misma función que para las patas en el aire.

γ : es el ángulo que forma la pata apoyada con la vertical, y ante la dificultad de medir el ángulo, ya que la pata es posible que deslice y un punto situado fijo en la vertical no nos proporcionaría resultados óptimos, se busca otra forma que consiste en medir la distancia horizontal entre el centro de masas y el punto inferior de la pata, y crear una función que calcule el arcoseno de ese valor dividido entre 200 que es la longitud existente entre estos dos puntos, de tal manera que nos queda:

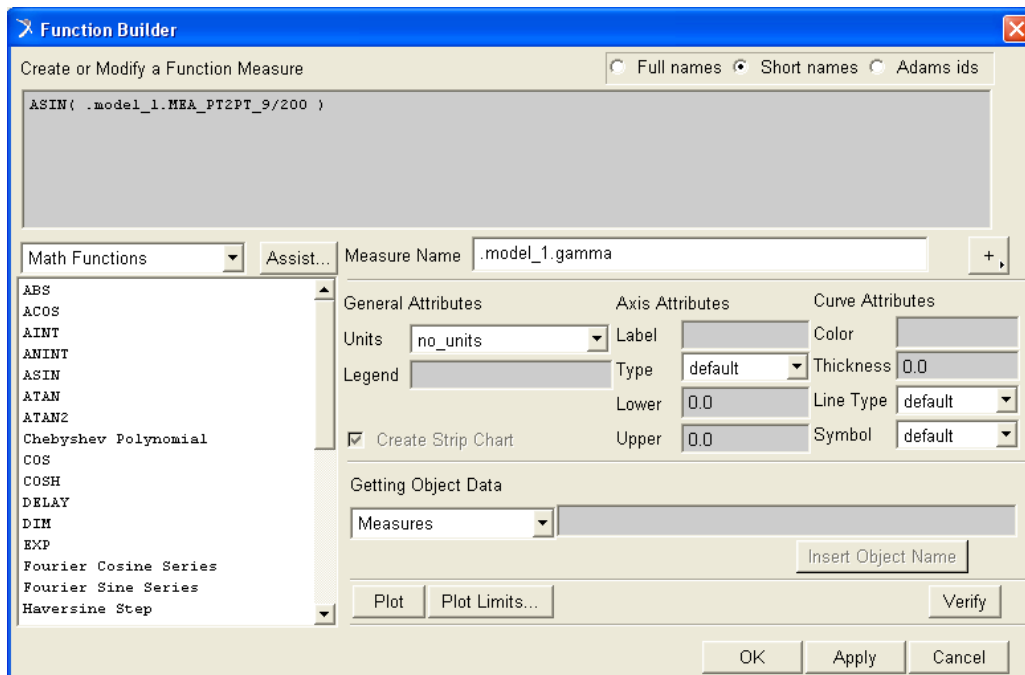


Figura 5.39 Creación de la función γ .

θ_s : es el mismo valor definido para las patas en el aire, con la diferencia de que en este caso, se divide entre dos, ya que el criterio a seguir es que la theta de referencia sea el doble que gamma.

$\dot{\gamma}$: este término define la velocidad angular de las patas apoyadas y se obtiene directamente de medirla en el elemento, tendremos únicamente en cuenta la componente Z.

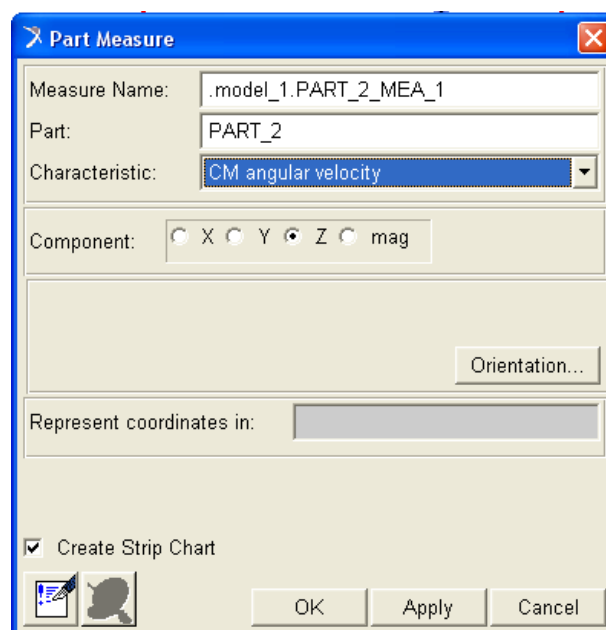


Figura 5.40 Medida de la velocidad angular $\dot{\gamma}$.

Renombramos esta medida como dergamma

$$\dot{\gamma} = \text{dergamma}$$

$m_{robot} g r_{robot x}$: al igual que en el caso de las otras patas este término varía con el tiempo durante la simulación pero no admite cambios de unas simulaciones a otras, por lo que introducimos el valor directamente en la fórmula.

$$m_{robot} = 2m_{leg} + m_{disco} = 4'92302 \text{ kg}$$

$$r_{robot x} = \frac{m_{leg} r_{leg apoyada x} + m_{disco} r_{disco x} + m_{leg} r_{leg movimiento x}}{m_{robot}}$$

$$= 331'001 \sin \gamma + 78'999 \sin(\theta - \gamma)$$

$$m_{robot} g r_{robot x} = 1'6 \cdot 10^7 \sin \gamma + 3813687'32 \sin(\theta - \gamma)$$

τ_f : es necesario compensar el par generado a las otras patas. Su valor se obtiene de la medida de este par en la componente z.

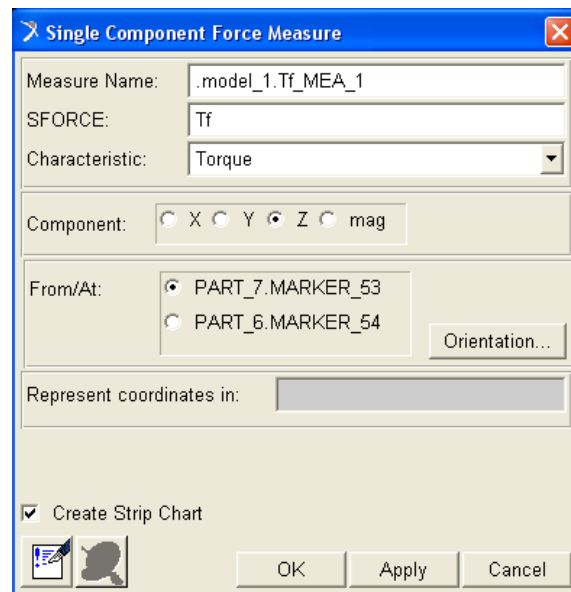


Figura 5.41 Medida del par Tf.

Teniendo en cuenta el mismo criterio de signos que para la pata en el aire (figura 5.37), e introduciendo los términos, nos queda la ecuación de control para la pata apoyada de la siguiente manera:

$$\begin{aligned} Th = & .model_1.Irobot * ((7/.model_1.ts)**2) * (-.model_1.gamma -.model_1.thetas/2) - \\ & .model_1.Irobot * (7/.model_1.ts) * .model_1.dergamma - \\ & 16000000 * SIN(.model_1.gamma) - 3813687.32 * SIN(.model_1.theta -.model_1.gamma) - \\ & .model_1.Tf_MEA_1 \end{aligned}$$

Siendo su forma introducida en Adams:

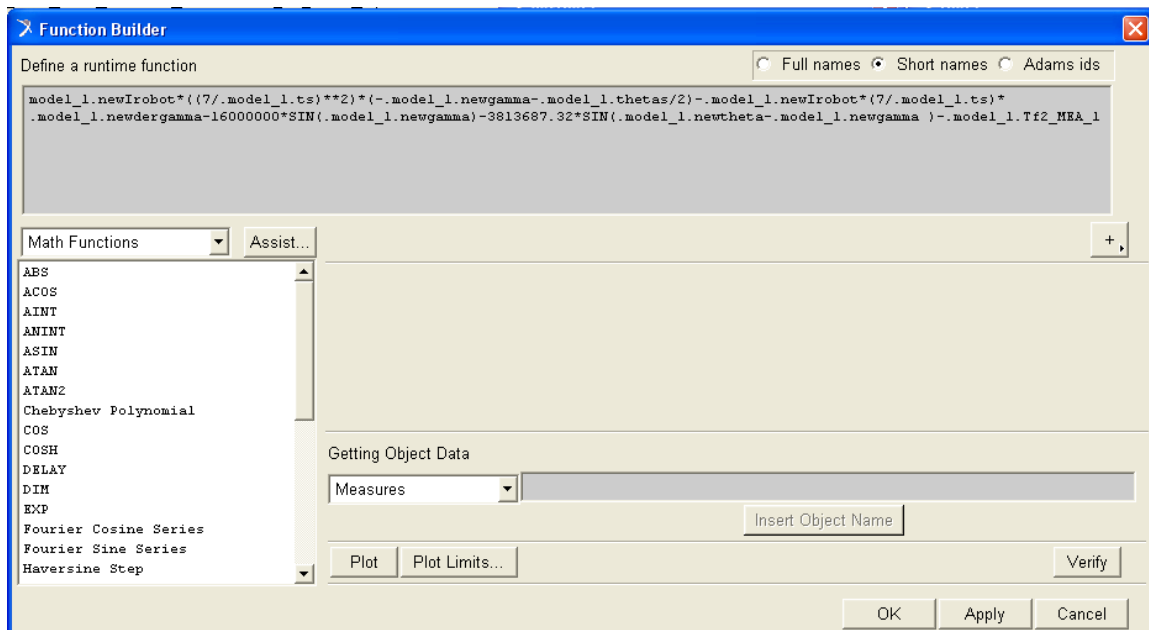


Figura 5.42 Creación de la función Th.

CAPÍTULO 6

Resultados de las simulaciones

6 RESULTADOS DE LAS SIMULACIONES

Una vez obtenido el modelo completo y sus algoritmos de control para proporcionar los correspondientes pares, se procede a comprobar su funcionamiento y robustez ante diferentes situaciones.

Se llevan a cabo simulaciones de un único paso, para que la toma de datos sea más sencilla, deteniendo el movimiento en el momento en que se produce el contacto entre superficies. Se tomarán medidas del periodo real empleado, ángulos alcanzados y la distancia recorrida por un punto guía de la pata para compararlo con los valores “objetivo” del algoritmo de control.

6.1 SIMULACIONES VARIANDO EL PERIODO DE CONTROL

El objetivo de este análisis es establecer el rango de periodos para el que el modelo con el algoritmo propuesto se comporta de forma correcta y establecer las desviaciones que se producen.

Para elaborar este análisis el ángulo que se desea alcanzar en el paso se mantendrá constante en 60° y el coeficiente de rozamiento con un valor elevado para que el deslizamiento no influya en este análisis. El comportamiento del modelo en función de esta propiedad se estudiará en apartados posteriores.

6.1.1 Periodo de 0'5 segundos

Se realiza en primer lugar una simulación con un periodo en el paso rápido, introduciendo los siguientes datos en la ecuación de control: $t_s = 0'5s$ $\theta_s = 1'0472rad$

La secuencia de paso en la simulación a través de imágenes de momentos puntuales se observa en la figura 6.

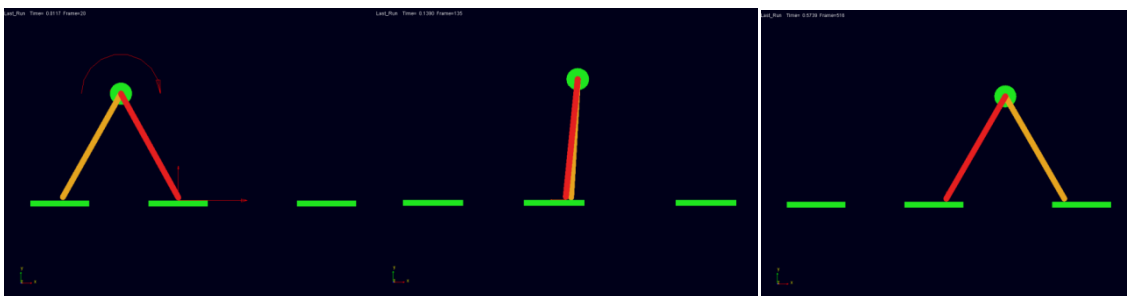


Figura 6.1 Simulación para $t_s = 0'5s$ $\theta_s = 1'0472rad$.

Ha sido necesario desplazar el escalón final hasta una posición más alejada, ya que la pata interaccionaba con este en su movimiento.

La simulación se detiene una vez completado el paso, siendo el periodo empleado y los ángulos alcanzados los que aparecen en la figura 6.2.

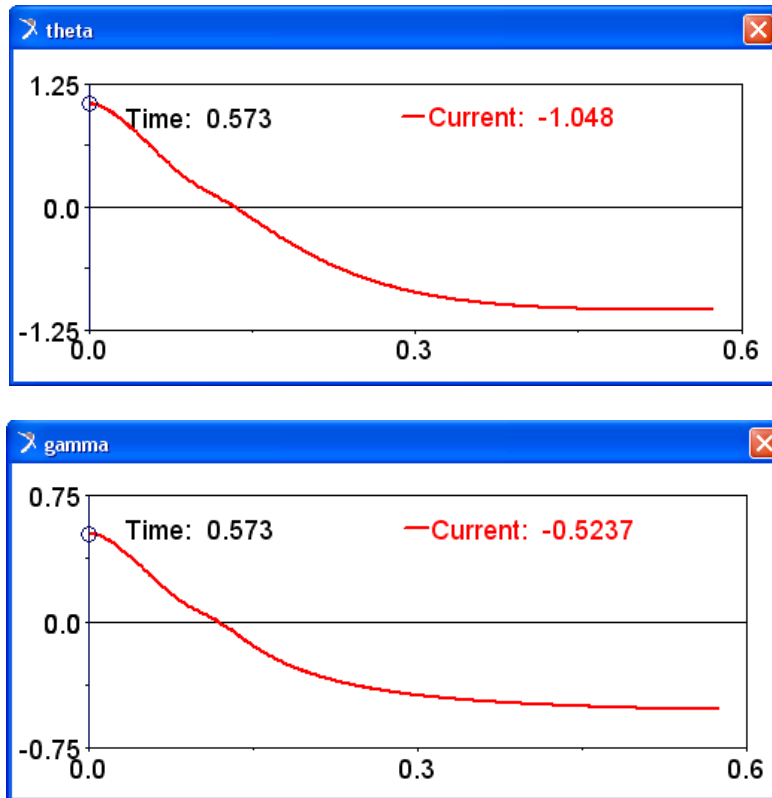


Figura 6.2 Resultados de ángulos para la simulación $t_s = 0'5s$ $\theta_s = 1'0472rad$.

Se observa en la primera imagen de la figura 6.1 como el par inicial generado es muy elevado ya que el movimiento es rápido.

Los ángulos alcanzados se aproximan al objetivo aunque el error cometido se tratará más detenidamente al final de este apartado.

6.1.2 Periodo de 0'75 segundos

En la segunda modificación se varía el valor de t_s que pasa a ser $t_s = 0'75s$, mientras que el valor de θ_s se mantiene constante $\theta_s = 1'0472rad$

Se realiza de igual manera la correspondiente simulación obteniendo los resultados gráficos de la evolución de los ángulos en función del tiempo.

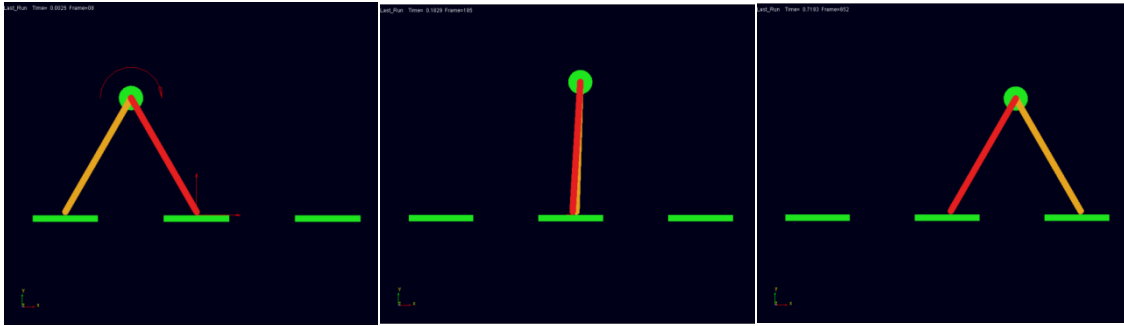


Figura 6.3 Simulación para $t_s = 0.75s$ $\theta_s = 1.0472rad$.

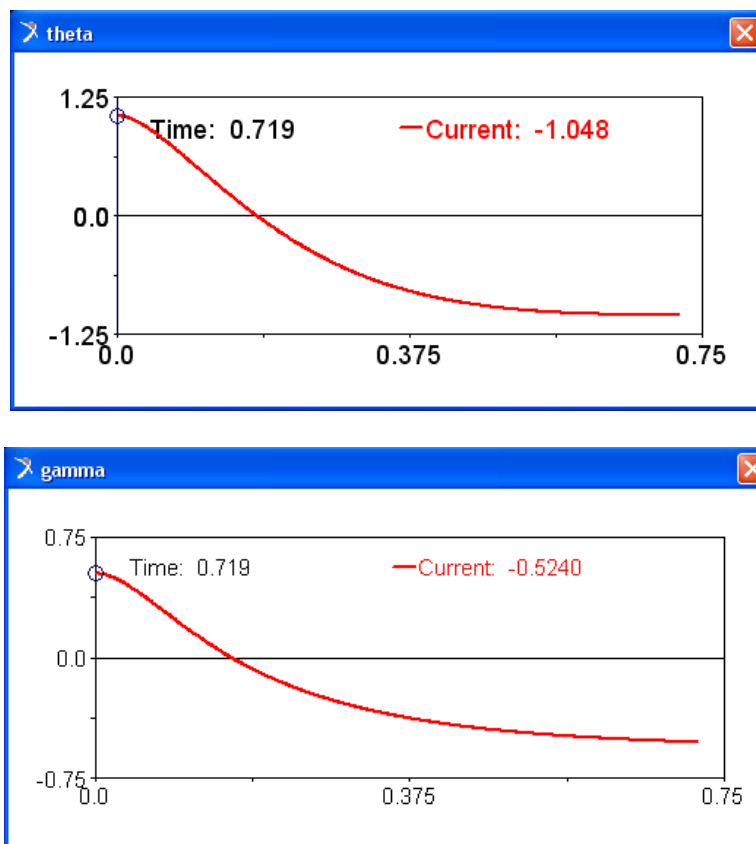


Figura 6.4 Resultados de ángulos para la simulación $t_s = 0.75s$ $\theta_s = 1.0472rad$.

Se observa como el par inicial disminuye, siendo el movimiento de avance más suave. Los ángulos alcanzados son similares a la primera simulación sin ser necesario en este caso desplazar el escalón.

6.1.3 Periodo de 1 segundo.

En la siguiente simulación t_s pasa a ser $t_s = 1s$, mientras que el valor de θ_s se mantiene constante $\theta_s = 1'0472rad$

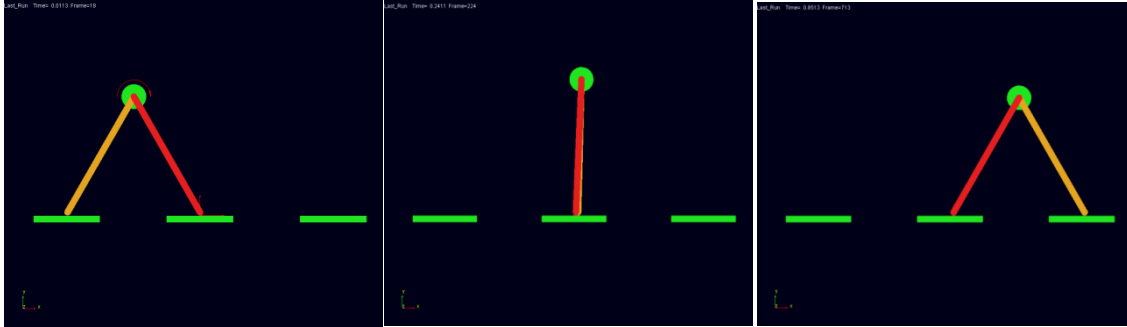


Figura 6.5 Simulación para $t_s = 1s$ $\theta_s = 1'0472rad$.

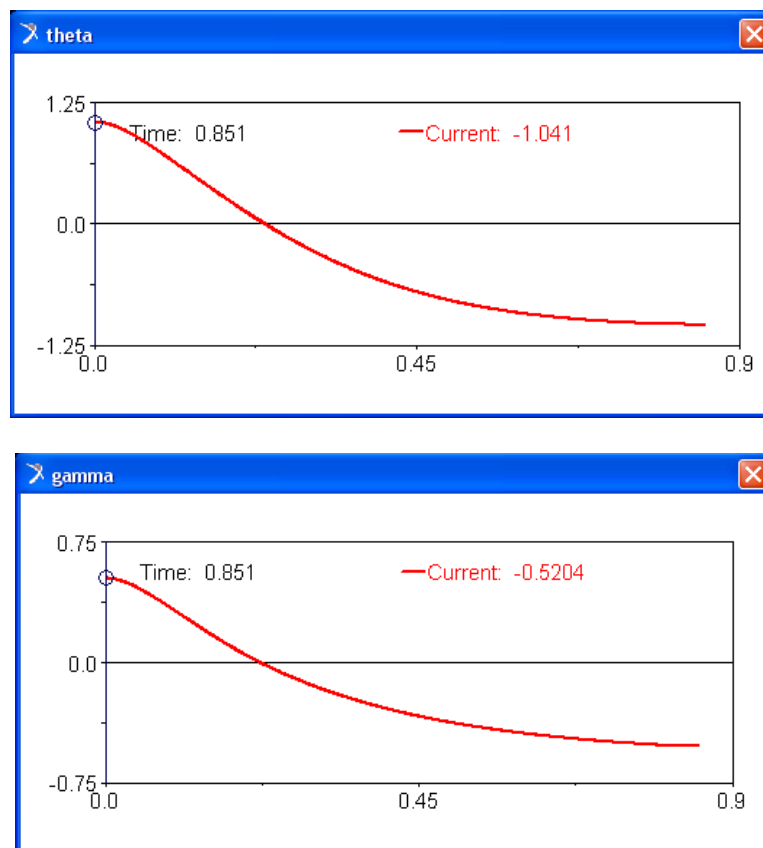


Figura 6.6 Resultados de ángulos para la simulación $t_s = 1s$ $\theta_s = 1'0472rad$.

En la simulación con periodo de 1 segundo se llega al contacto con la superficie del escalón antes de haber alcanzado el ángulo final deseado, siendo el tiempo empleado además menor del introducido a diferencia de las anteriores simulaciones.

6.1.4 Periodo de 1'25 segundos.

Como en las simulaciones anteriores, manteniendo el mismo θ_s , se modifica el periodo, en este caso a $t_s = 1'25s$.

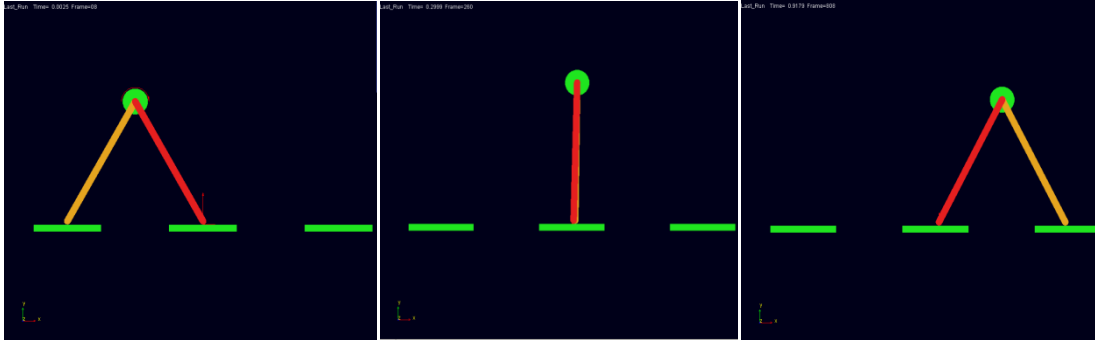


Figura 6.7 Simulación para $t_s = 1'25s$ $\theta_s = 1'0472rad$.

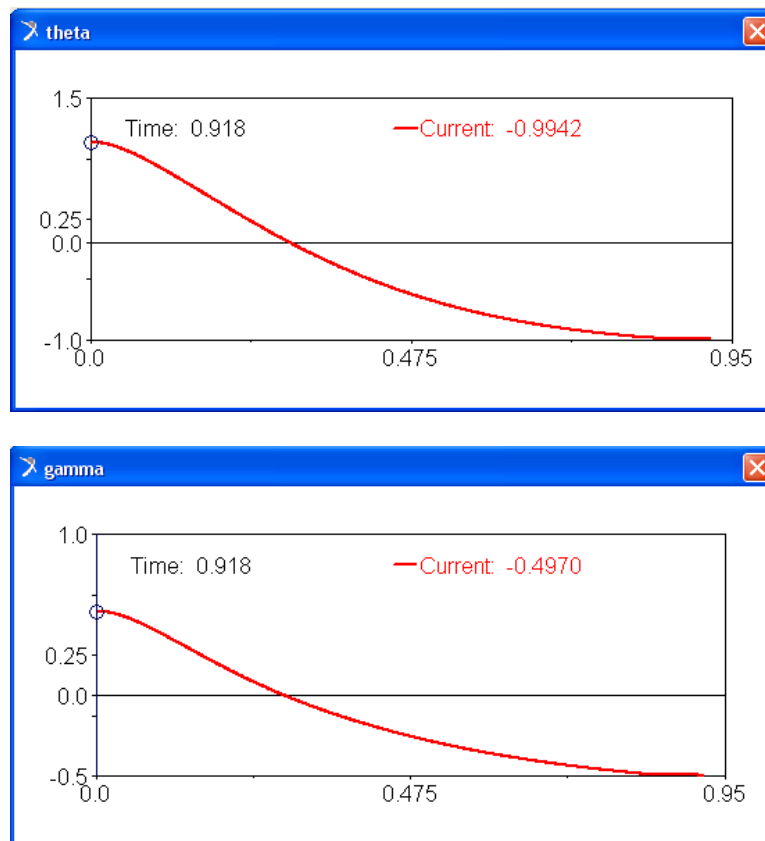


Figura 6.8 Resultados de ángulos para la simulación $t_s = 1'25s$ y $\theta_s = 1'0472rad$.

Siguiendo la tendencia observada anteriormente el contacto se produce con un tiempo y ángulos inferiores a los valores previstos.

6.1.5 Periodo de 1'5 segundos.

Como en las simulaciones anteriores, manteniendo constante el valor θ_s , se modifica el periodo, en este caso a $t_s = 1'5s$.

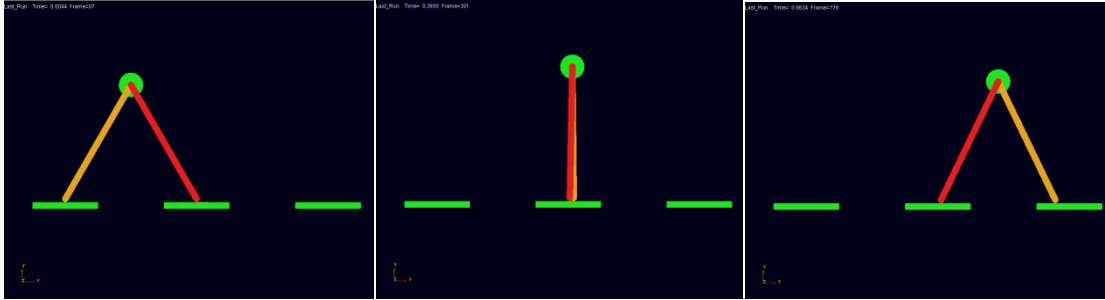


Figura 6.9 Simulación para $t_s = 1'5s$ $\theta_s = 1'0472rad$.

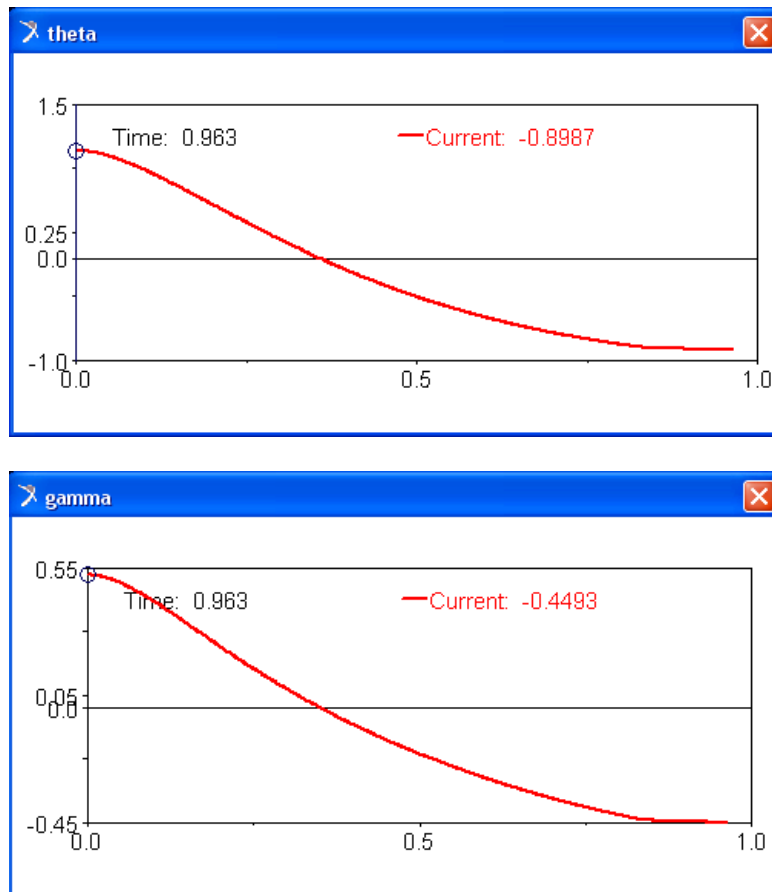


Figura 6.10 Resultados de ángulos para la simulación $t_s = 1'5s$ y $\theta_s = 1'0472rad$.

6.1.6 Periodo de 2 segundos.

Para observar con más claridad cómo funciona el robot se eleva el periodo hasta los 2 segundos.

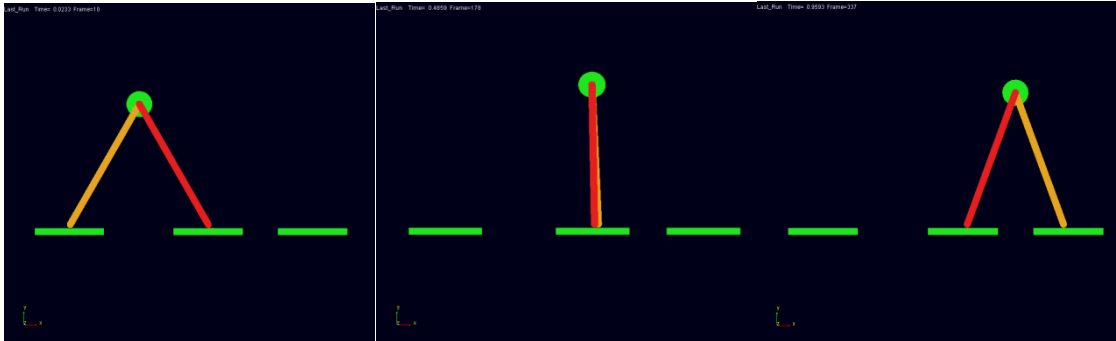


Figura 6.11 Simulación para $t_s = 2s$ $\theta_s = 1'0472rad$.

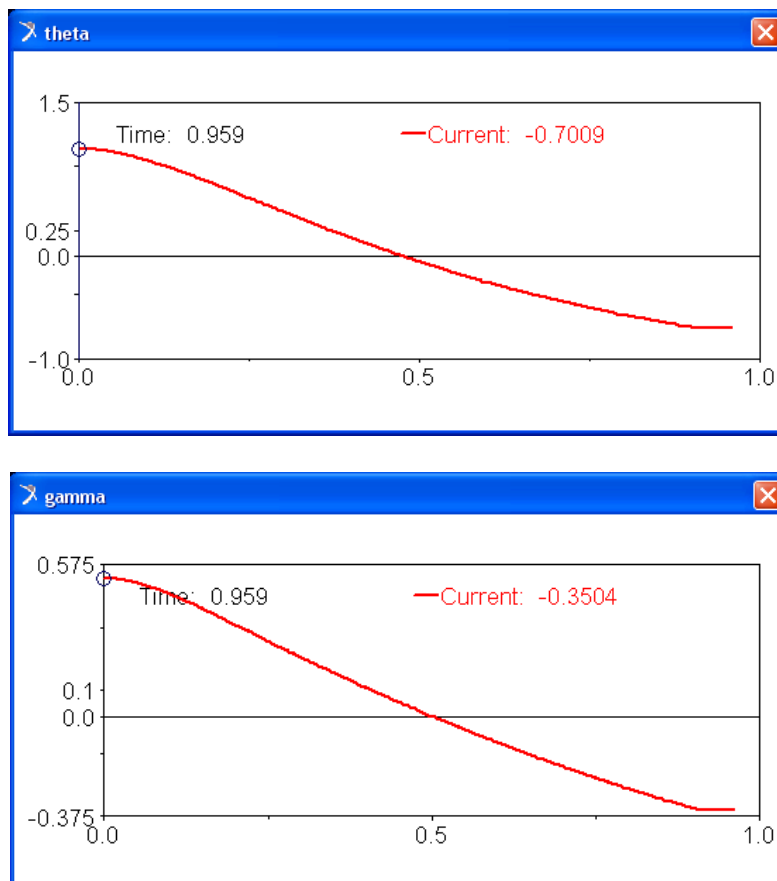


Figura 6.12 Resultados de ángulos para la simulación $t_s = 2s$ y $\theta_s = 1'0472rad$.

De forma clara se ve en este caso como el movimiento no se completa de manera correcta.

6.1.7 Resumen de los datos obtenidos en las simulaciones.

Una vez realizadas las diferentes simulaciones recogemos todos los datos en la siguiente tabla para un análisis más claro. Se incluye además las medidas realizadas sobre la distancia recorrida de un punto de la pata tomada como referencia que no se expresaron anteriormente.

Periodo de control (s)	Periodo obtenido simulación (s)	Theta control (rad)	Theta obtenido simulación (rad)	Gamma de control (rad)	Gamma obtenido simulación (rad)	Recorrido de control (mm)	Recorrido obtenido simulación (mm)
0'5	0'573	1'0472	1'048	0'5236	0'5237	800	848'5
0'75	0'719	1'0472	1'048	0'5236	0'5240	800	811
1	0'851	1'0472	1'041	0'5236	0'5204	800	808'2
1'25	0'918	1'0472	0'9942	0'5236	0'4970	800	791'1
1'5	0'963	1'0472	0'8987	0'5236	0'4493	800	757'9
2	0'959	1'0472	0'7009	0'5236	0'3504	800	687'2

Tabla 6.1 Resultados de simulaciones con cambios de periodos.

Se calculan los errores cometidos respecto de los datos esperados, representándolos en sus correspondientes gráficas.

$$\varepsilon(\%) = \frac{|Valor\ de\ control - Valor\ obtenido|}{|Valor\ de\ control|} 100$$

Periodo deseado (s)	Error periodo (%)	Error theta (%)	Error gamma (%)	Error recorrido (%)
0'5	14'6	0'076	0'019	6'06
0'75	4'13	0'076	0'076	1'375
1	14'9	0'592	0'61	1'0025
1'25	26'56	5'06	5'08	1'11
1'5	35'8	15'15	14'19	5'2625
2	52'05	33'07	33'07	14'1

Tabla 6.2 Cálculo de errores en simulaciones con cambios de periodos.

- Periodos

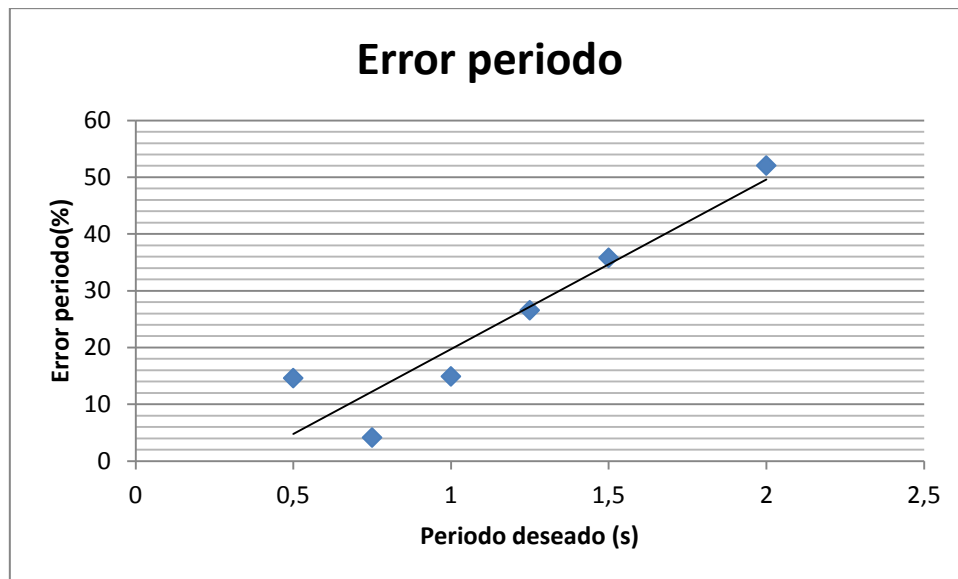


Figura 6.13 Representación gráfica de los errores de periodos.

Representando el error porcentual resultante de la comparación de los periodos reales obtenidos en las simulaciones frente a los periodos de control esperados, se puede observar como la tendencia es casi lineal. Esto significa que a medida que se aumenta el periodo deseado para el paso, más error se produce y por tanto menos precisión en el movimiento. Es necesario resaltar que esta regla no se cumple para periodos muy pequeños, ya que el movimiento comienza con pares muy elevados lo que provoca que el movimiento este menos controlado.

- Ángulos

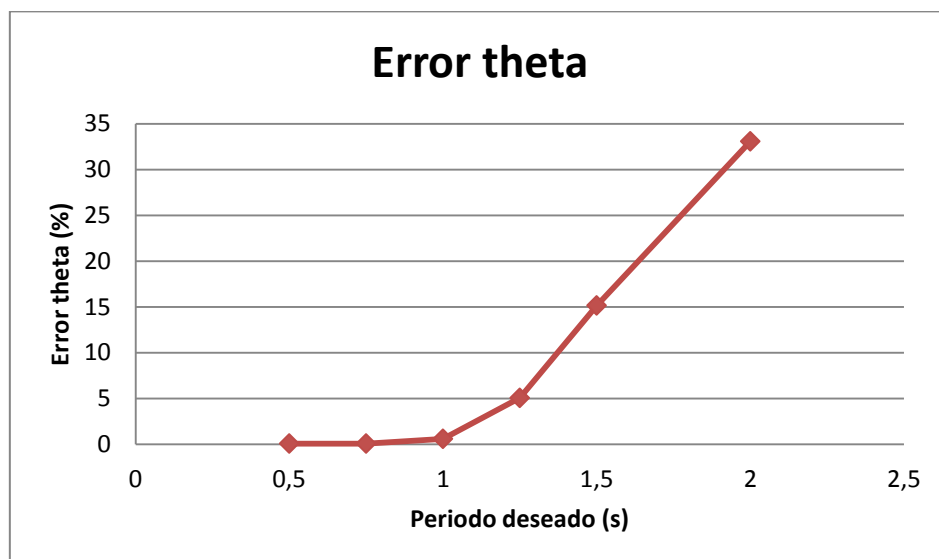


Figura 6.14 Representación gráfica de los errores de θ .

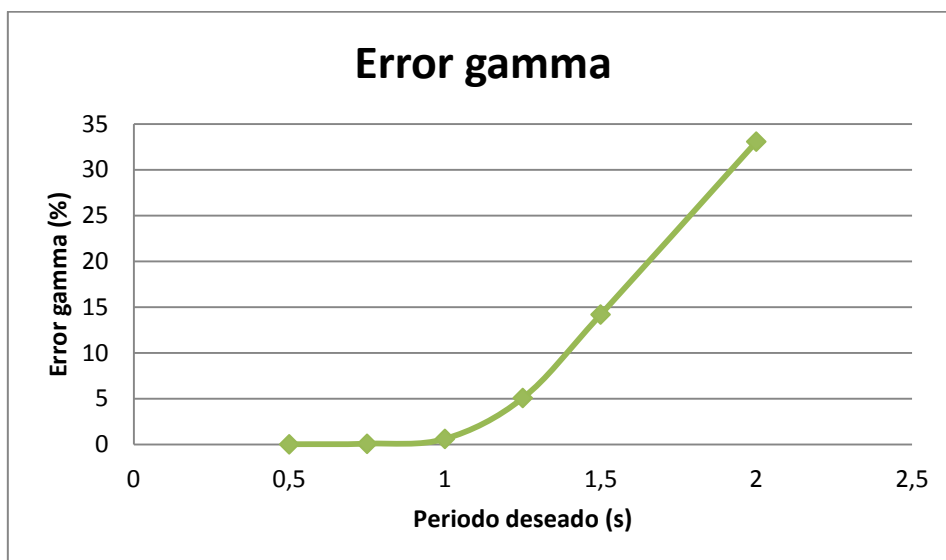


Figura 6.15 Representación gráfica de los errores de γ .

Los errores de theta y gamma pueden ser tratados simultáneamente ya que como se observa siguen exactamente la misma curva. Esto es debido a que según las ecuaciones de control el objetivo es que $\gamma = \frac{\theta}{2}$ por lo tanto los errores cometidos serán iguales.

El error en este caso aumenta de una forma mucho mayor cuanto mayor es el periodo del movimiento. Esto lleva a la conclusión de que el error cometido en el periodo influye de manera considerable en la amplitud de paso.

- Distancias recorridas.



Figura 6.16 Representación gráfica de los errores de distancias.

Para periodos grandes o muy pequeños el error en la distancia recorrida por la pata es mayor. Esta idea concuerda con las deducciones de los apartados anteriores, si los errores de periodo y ángulos son mayores en ese rango consecuentemente también lo será el de la distancia que depende directamente de las otras propiedades.

Si en lugar de tratar el error en valor absoluto se observan las tablas de resultados se comprueba que para periodos pequeños la amplitud del paso es mayor que la deseada, mientras que para periodos largos la amplitud es menor que el objetivo.

6.2 SIMULACIONES VARIANDO LA AMPLITUD DE CONTROL EN EL PASO.

Una vez estudiado el comportamiento del modelo frente a los cambios de periodo, el siguiente paso es comprobar como lo hace ante variaciones en la amplitud del paso.

En las simulaciones se parte de una posición inicial de 60° , pero el siguiente paso se efectuará con una amplitud diferente, manteniendo en todos los casos el periodo constante. A tenor de los resultados anteriores escogemos el periodo de 0.75 segundos para las simulaciones, ya que sus errores son los más bajos.

El rango de amplitudes en el que se trabaja será de 15 a 90° .

6.2.1 Amplitud de 15°

Se realiza en primer lugar una simulación con $t_s = 0.75s$ y $\theta_s = 0.2618rad$. Es necesario juntar el escalón final ya que la distancia recorrida por la pata es menor que en las simulaciones anteriores.

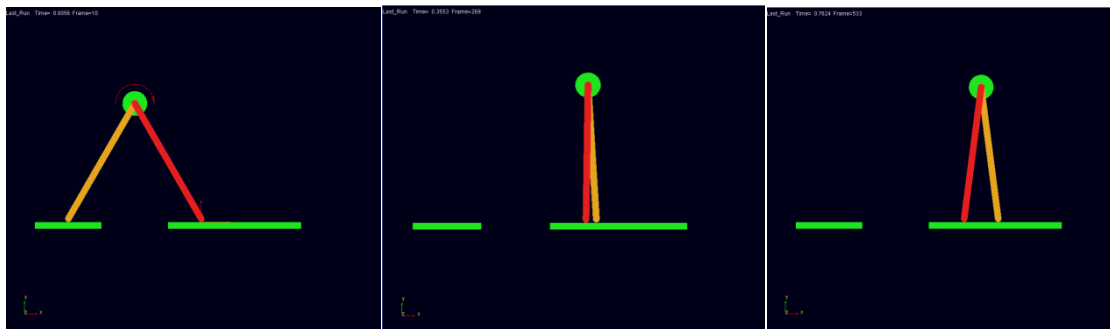


Figura 6.17 Simulación para $t_s = 0.75s$ $\theta_s = 0.2618rad$.

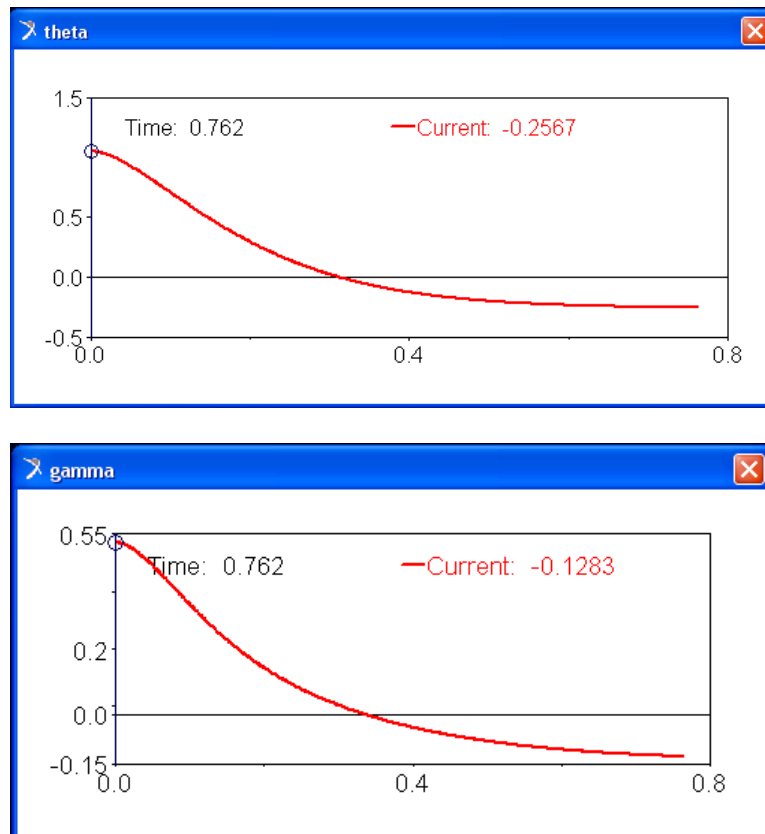


Figura 6.18 Resultados de ángulos para la simulación $t_s = 0'75s$ y $\theta_s = 0'2618rad$.

Los resultados obtenidos a priori parecen buenos, desarrollándose la simulación sin problemas en el movimiento. Los errores cometidos se tratarán al final del presente apartado.

6.2.2 Amplitud de 30°

Para el segundo ensayo se mantiene constante el periodo con $t_s = 0'75s$, modificando el ángulo $\theta_s = 0'5236rad$. La plataforma final permanece en una posición más cercana, para facilitar que la pata alcance su objetivo.

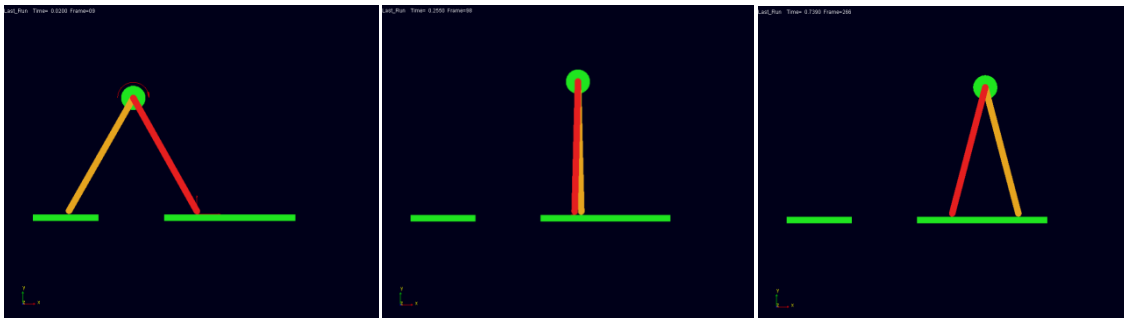


Figura 6.19 Simulación para $t_s = 0'75s$ $\theta_s = 0'5236rad$.

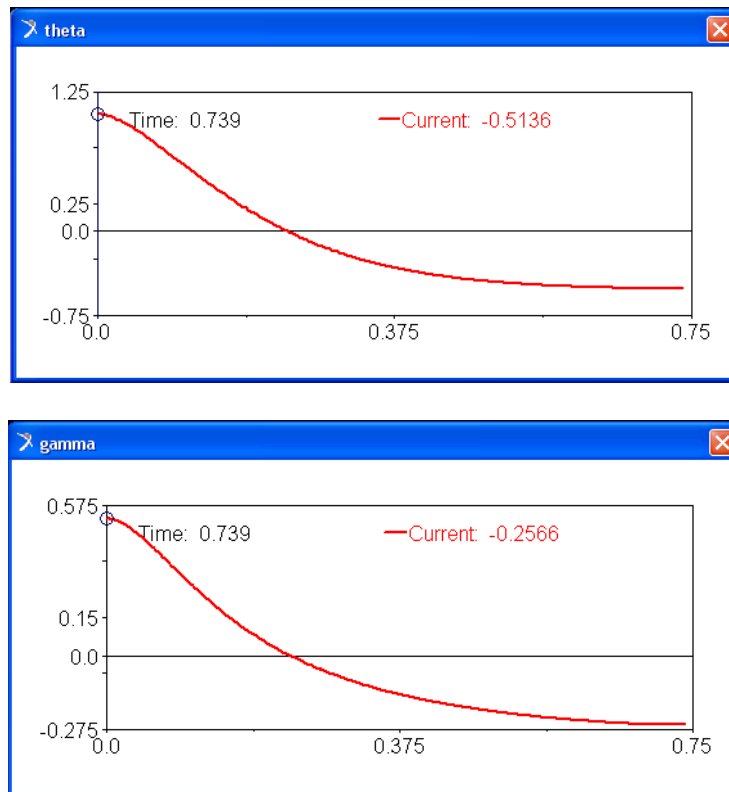


Figura 6.20 Resultados de ángulos para la simulación $t_s = 0'75s$ y $\theta_s = 0'5236rad$.

Realizada la simulación se observa un movimiento de avance controlado sin producirse interferencias con los escalones, problema que podría haber surgido por la situación de estos. Los resultados de ángulos obtenidos se aproximan al objetivo.

6.2.3 Amplitud de 45°

Se realiza un nuevo ensayo con $t_s = 0'75s$ y $\theta_s = 0'7854rad$. La plataforma final vuelve a estar situada en su posición inicial, ya que la longitud que alcanza la pata es mayor.

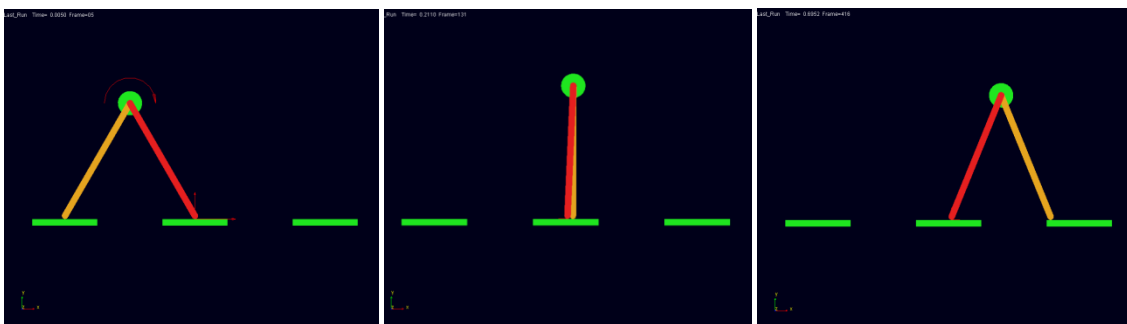


Figura 6.21 Simulación para $t_s = 0'75s$ $\theta_s = 0'7854rad$.

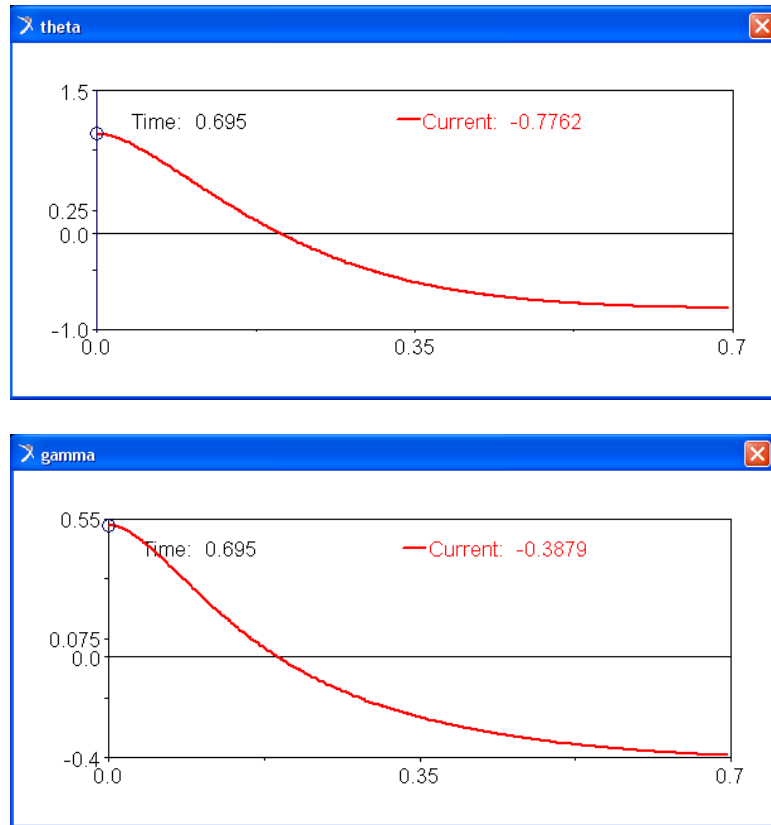


Figura 6.22 Resultados de ángulos para la simulación $t_s = 0'75s$ y $\theta_s = 0'7854rad$.

Se puede observar ya en esta simulación como el par inicial con el que se comienza el movimiento es mayor que en los casos anteriores, concuerda con lo esperado ya que la fuerza necesaria tendrá que ser mayor si las patas tienen que alcanzar un punto final más alejado en el mismo tiempo.

6.2.4 Amplitud de 60°

En este caso $\theta_s = 1'0472rad$ y $t_s = 0'75s$, por lo tanto se trata del mismo ensayo realizado en el apartado de cambios de periodo.

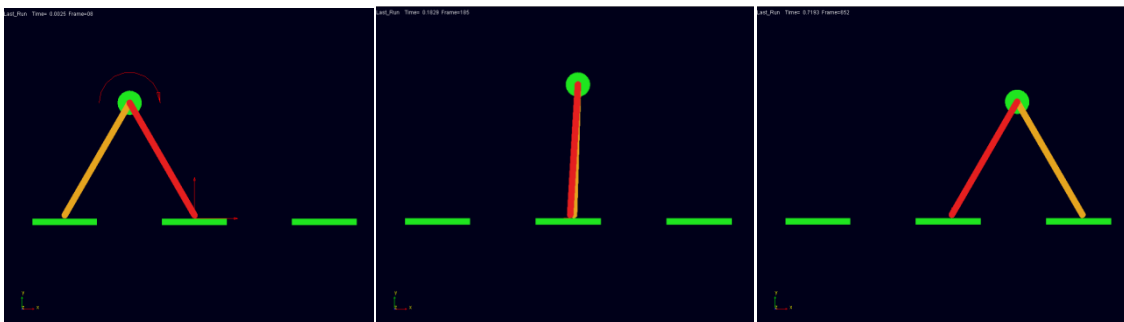


Figura 6.23 Simulación para $t_s = 0'75s$ $\theta_s = 1'0472rad$.

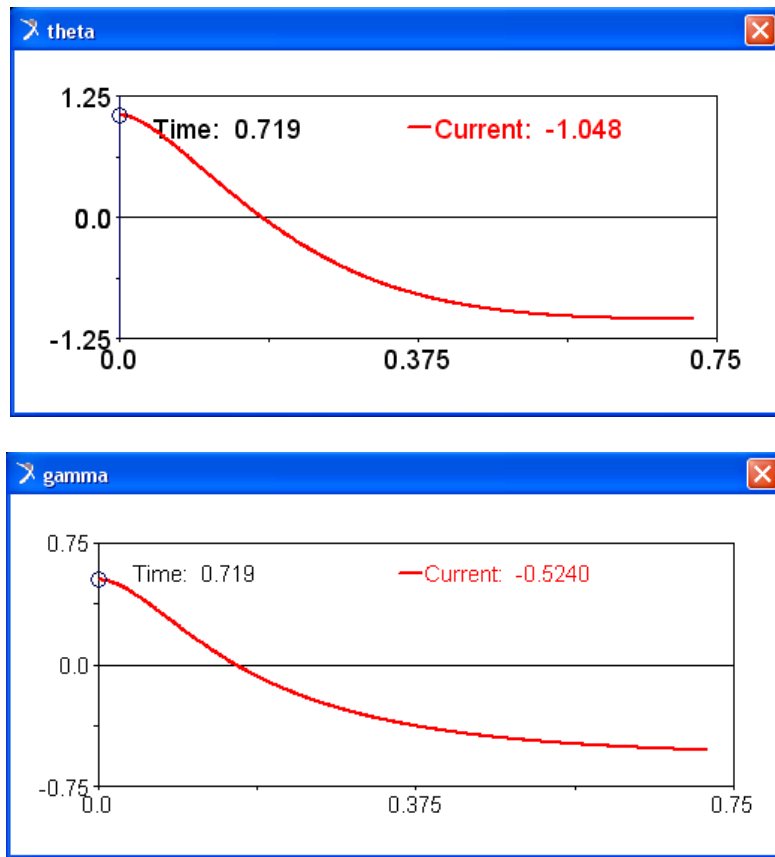


Figura 6.24 Resultados de ángulos para la simulación $t_s = 0'75s$ y $\theta_s = 1'0472rad$.

6.2.5 Amplitud de 75°

Se sigue ampliando la amplitud final entre patas, en este caso hasta $\theta_s = 1'309rad$

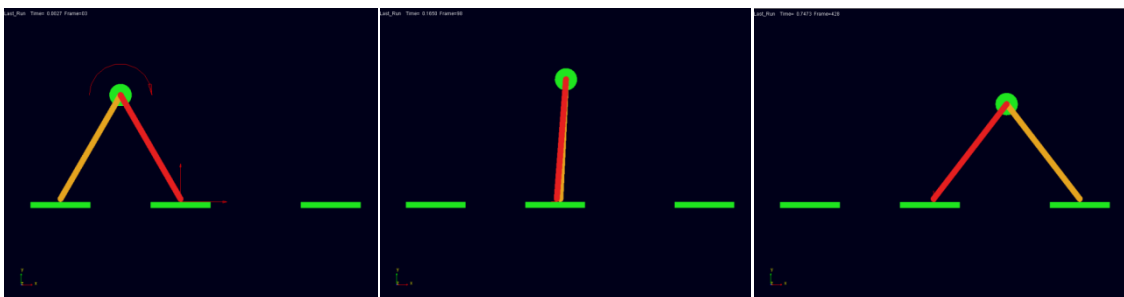


Figura 6.25 Simulación para $t_s = 0'75s$ $\theta_s = 1'309rad$.

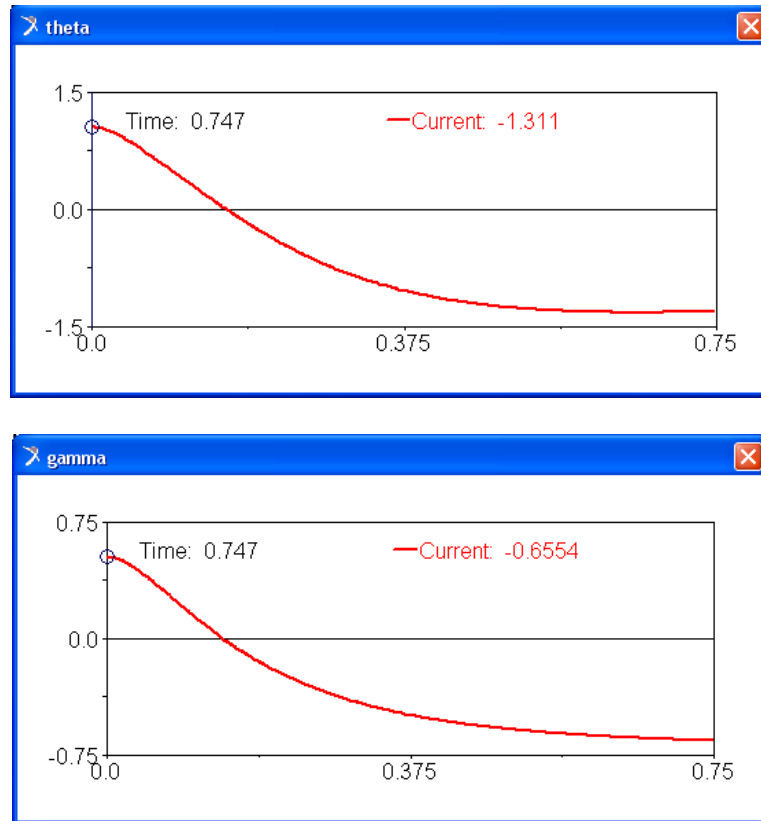


Figura 6.26 Resultados de ángulos para la simulación $t_s = 0.75s$ y $\theta_s = 1.309rad$.

Se desplaza la plataforma final hasta una posición más alejada para facilitar el movimiento de las patas sin que se produzcan interacciones, obteniendo buenos resultados en la simulación.

6.2.6 Amplitud de 90°

Como último ensayo respecto a lo que amplitudes se refiere, tomamos un valor de $\theta_s = 1.5708rad$, cubriendo así un espectro amplio entre amplitudes pequeñas y grandes.

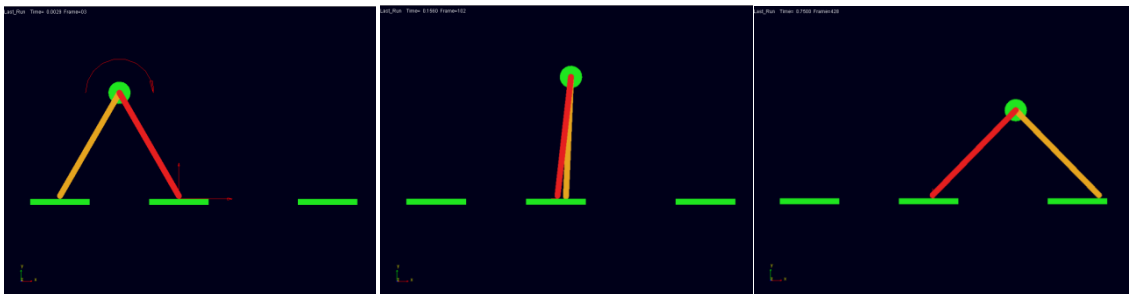


Figura 6.27 Simulación para $t_s = 0.75s$ $\theta_s = 1.5708rad$.

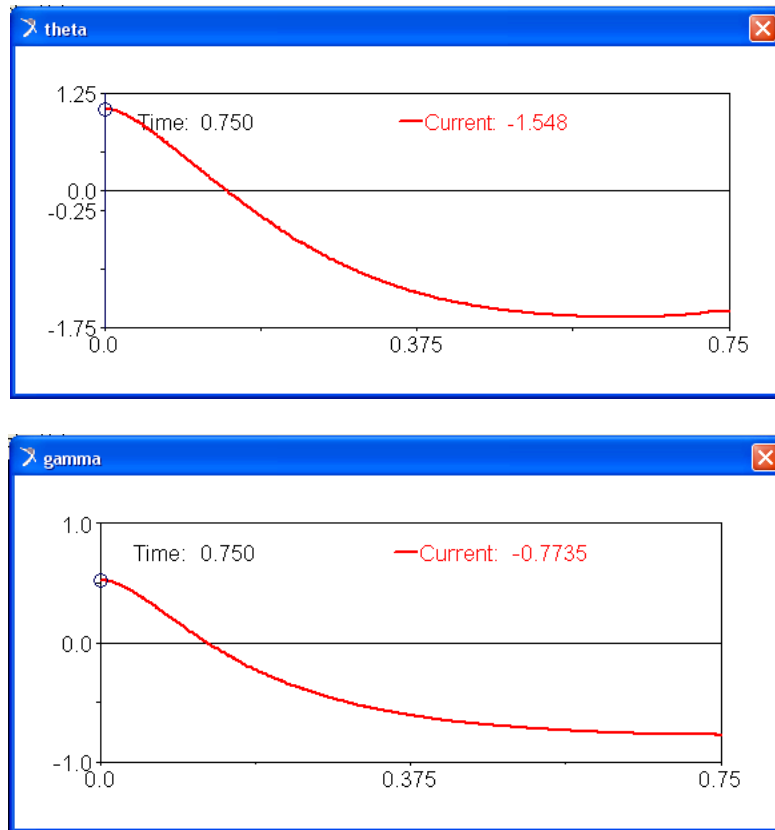


Figura 6.28 Resultados de ángulos para la simulación $t_s = 0.75s$ y $\theta_s = 1.5708rad$.

La plataforma final se mantiene en la posición del ensayo anterior, distancia suficiente para el correcto desarrollo de la simulación.

6.2.7 Resumen de simulaciones

Al igual que en las simulaciones con cambio de periodo, una vez realizadas las diferentes simulaciones recogemos todos los datos en una tabla calculando los correspondientes errores. Se incluye además la distancia de referencia recorrida por la pata en comparación con lo esperado.

Periodo de control (s)	Periodo obtenido simulación (s)	Theta de control (rad)	Theta obtenido simulación (rad)	Gamma de control (rad)	Gamma obtenido simulación (rad)	Recorrido de control (mm)	Recorrido obtenido simulación (mm)
0'75	0'762	0'2618	0'2567	0'1309	0'1283	504'4	509'3
0'75	0'739	0'5236	0'5136	0'2618	0'2566	607'1	611'4
0'75	0'695	0'7854	0'7762	0'3927	0'3879	706'1	712'1
0'75	0'719	1'0472	1'048	0'5236	0'5240	800'0	811
0'75	0'747	1'3090	1'311	0'6545	0'6554	887'0	900'9
0'75	0'750	1'5707	1'548	0'7854	0'7735	965'7	973'8

Tabla 6.3 Resultados de simulaciones con cambios de amplitud.

Se calcula los errores cometidos respecto de los datos esperados, representándolos en sus correspondientes gráficas.

$$\varepsilon(\%) = \frac{|Valor\ de\ control - Valor\ obtenido|}{|Valor\ de\ control|} 100$$

Theta deseado (rad)	Error periodo (%)	Error theta (%)	Error gamma (%)	Error recorrido (%)
0'2618	1'6	1'948	1'9862	0'971
0'5236	1'467	1'910	1'9862	0'708
0'7854	7'33	1'171	1'2223	0'850
1'0472	4'13	0'0764	0'0764	1'375
1'3090	0'3	0'1528	0'1375	1'567
1'5707	0	1'4452	1'5151	0'839

Tabla 6.4 Cálculo de errores en simulaciones con cambios de amplitud.

- Periodos

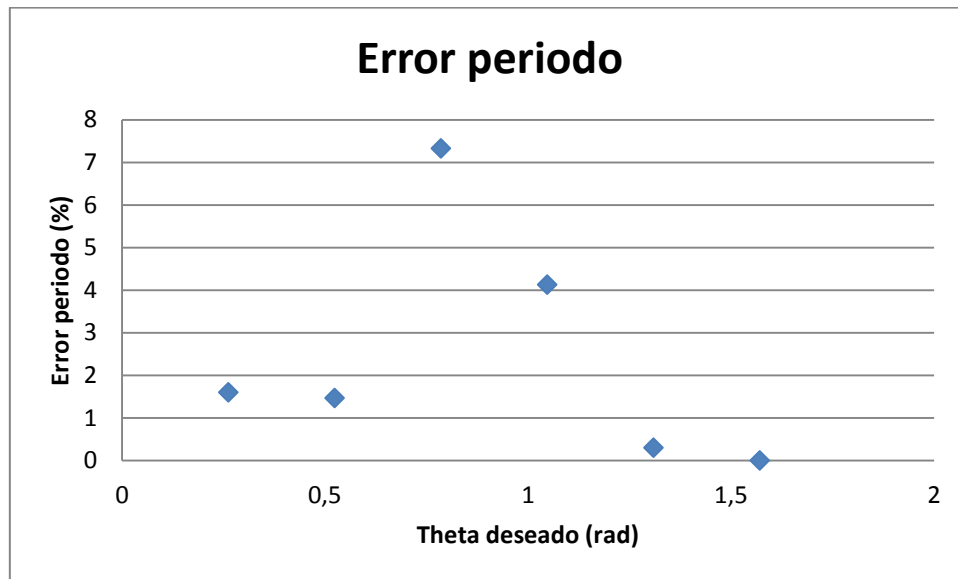


Figura 6.29 Representación gráfica de los errores de periodos.

Graficando los errores existentes en los periodos de las diferentes simulaciones se puede observar que estos no siguen una tendencia dependiente del ángulo. Aunque existen algunos valores por encima del resto su magnitud es mucho menor en todos los casos que los errores que se observaban en las simulaciones con cambios de periodo.

- Ángulos

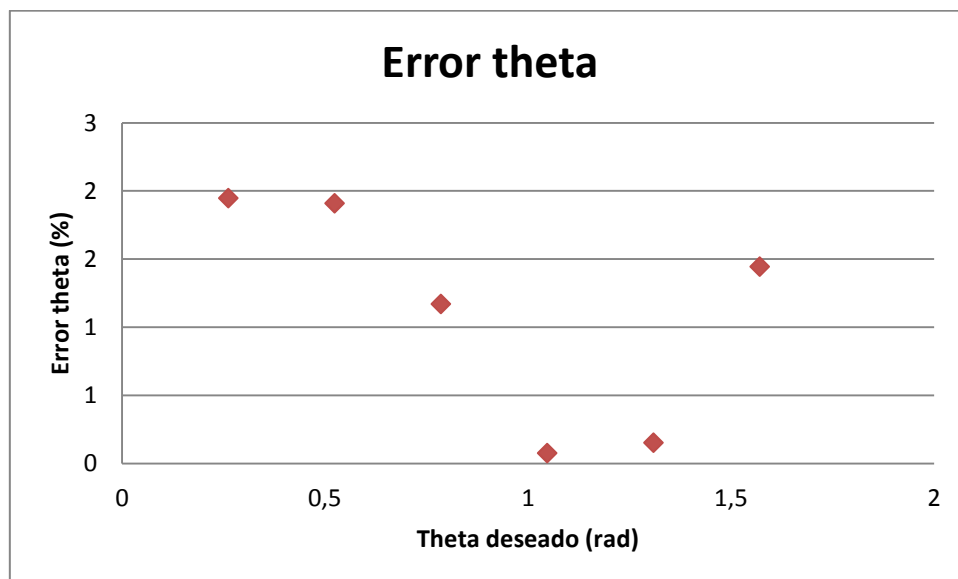


Figura 6.30 Representación gráfica de los errores de θ .

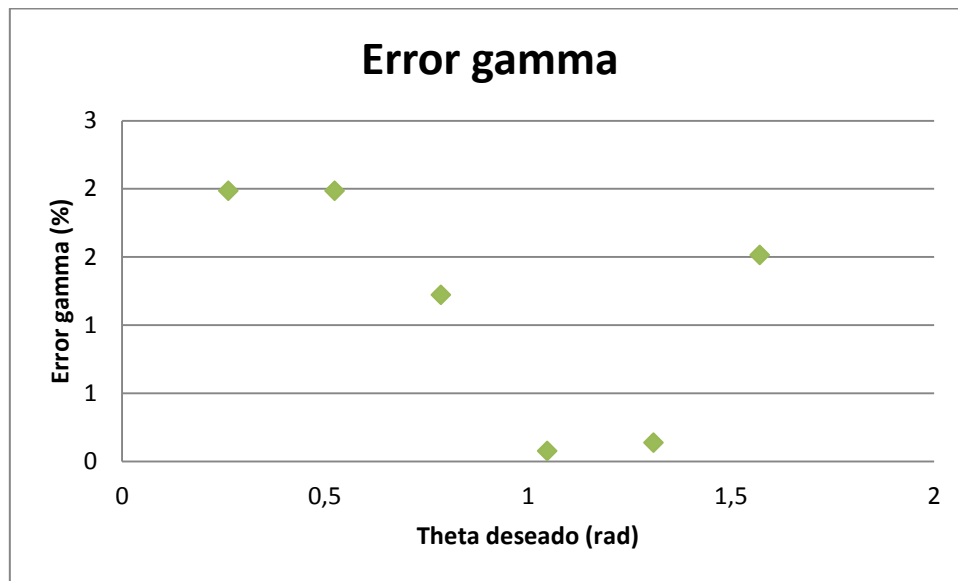


Figura 6.31 Representación gráfica de los errores de θ .

Tratando las gráficas de los errores cometidos en los ángulos se puede destacar la gran capacidad que tiene el modelo creado para adaptarse a los cambios de ángulos. Los datos no siguen una tendencia y se encuentran todos por debajo del 2%.

- Distancias recorridas.

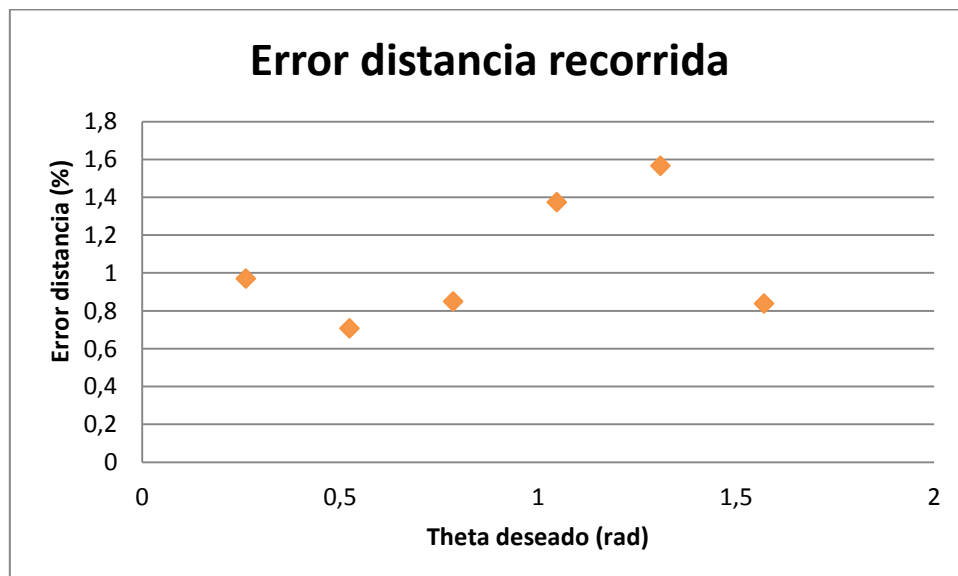


Figura 6.32 Representación gráfica de los errores de distancias.

En el caso de las distancias recorridas los valores son muy parejos y observando los datos en la tabla se puede ver que las distancias recorridas son siempre mayores a las esperadas aunque en un porcentaje prácticamente despreciable.

6.3 SIMULACIONES VARIANDO EL COEFICIENTE DE ROZAMIENTO

Un aspecto modificable y que puede tener mucha importancia a la hora de realizar movimientos es el coeficiente de rozamiento existente entre las patas y el suelo.

Escogemos una combinación de ángulo y periodo que en los ensayos anteriores se demostró que realizaba el movimiento de forma correcta y con errores despreciables. Se escoge un periodo $t_s = 0'75s$ y un ángulo $\theta_s = 1'0472rad$.

Se incrementa en este apartado el número de imágenes por simulación, ya que es igual de interesante conocer lo que sucede en los pasos intermedios que su posición final.

Se han realizado simulaciones de un paso del bípedo variando el periodo y el coeficiente de rozamiento entre las patas y el suelo. Los resultados indican que cuanto menor es el periodo de control, mayor tiene que ser el coeficiente de rozamiento para que la pata de apoyo no deslice apreciablemente.

Así, por ejemplo para un periodo de 0'75 segundos, coeficientes de rozamiento inferiores a $\mu=1'5$ producen deslizamientos mayores del 5 % de la distancia total recorrida. Para un periodo de control de 1 segundo, el valor límite del coeficiente de rozamiento se reduce a $\mu=0'9$ y para un periodo de 1.25 segundos, a $\mu=0'7$.

Se muestran algunos ejemplos que corroboran las ideas planteadas:

- Para $t_s = 0'75s$



Figura 6.33 Simulación para $t_s = 0'75s$ y $\mu=1$

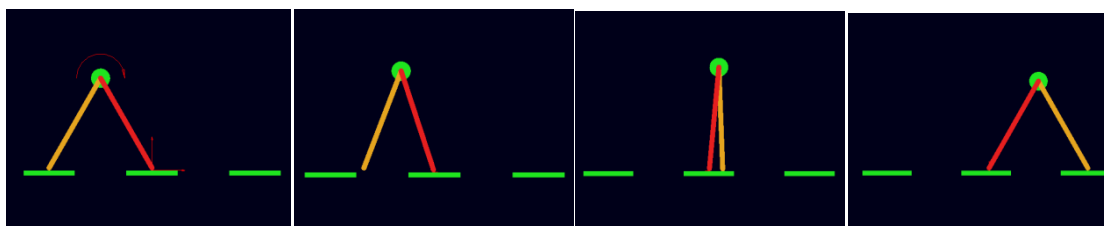


Figura 6.34 Simulación para $t_s = 0'75s$ y $\mu=1'5$

- Para $t_s = 1s$

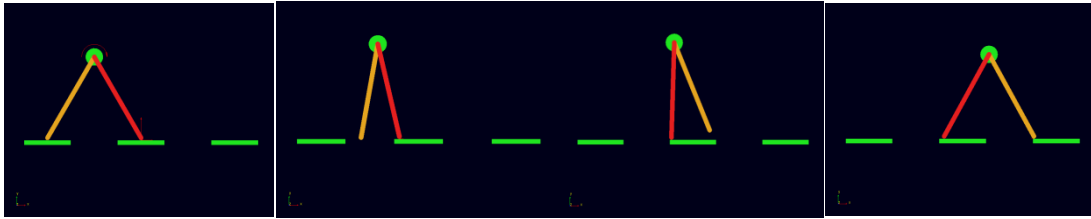


Figura 6.35 Simulación para $t_s = 1s$ y $\mu=0'5$

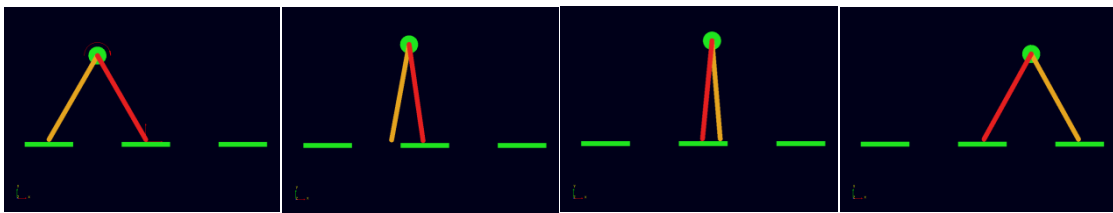


Figura 6.36 Simulación para $t_s = 1s$ y $\mu=0'9$

- Para $t_s = 1'25s$

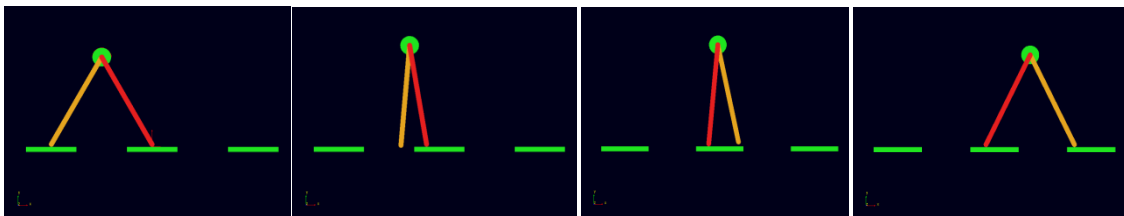


Figura 6.37 Simulación para $t_s = 1'25s$ y $\mu=0'5$

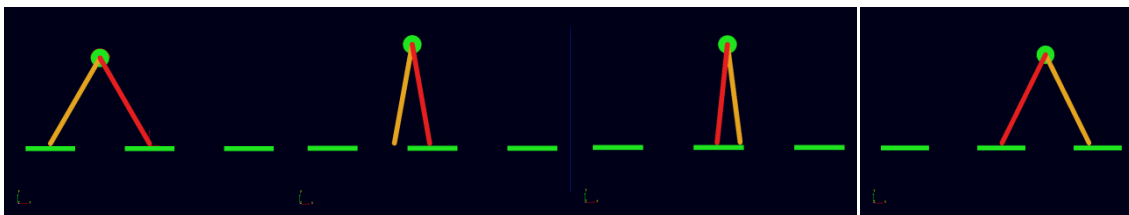


Figura 6.38 Simulación para $t_s = 1'25s$ y $\mu=0'7$

Se puede elegir por lo tanto entre dos alternativas: valores de periodo bajos, que desarrollan el movimiento con errores menores, aplicando coeficientes de rozamiento altos; o emplear periodos mayores que necesitan de coeficientes de rozamiento más bajos, pero que desarrollan el paso con porcentajes más elevados de error.

6.4 SIMULACIÓN CON DOBLE PASO

6.4.1 Introducción de nuevos pares

En todos los ensayos anteriores únicamente se realizaba un paso en las simulaciones porque el objetivo era comprobar cómo respondía el algoritmo de control a variaciones del periodo y la amplitud de dicho paso. Es interesante, una vez conocidos estos aspectos, encadenar pasos de una manera repetida simulando el caminar del robot.

Para realizar las simulaciones se utilizarán las condiciones que resultaron más favorables a la hora de realizar el movimiento de manera que los errores sean los menores posibles, pero siempre manteniendo unas condiciones reales de ensayo.

Se escogen por lo tanto unas condiciones de periodo 1 segundo, una amplitud de 60° entre las patas para realizar el movimiento y un coeficiente de rozamiento entre la pata y el suelo de 0.9.

Ante la dificultad de que las ecuaciones de control que gobiernan cada pata se intercambien entre sí para ejecutar el nuevo paso, se crean otras nuevas ecuaciones similares a las anteriores pero en las que se ha sustituido los componentes que las formaban por otros nuevos con las mismas estructuras de medida pero adaptadas al segundo paso. Actuarán las primeras hasta completar el primer paso y acto seguido se desconectarán pasando el robot a ser gobernado por las segundas.

En las figuras 6.39 y 6.40 se presentan las ventanas de Adams donde se introducen los códigos de control en los que se contempla el intercambio de funciones entre patas

Nueva pata en el aire

```
Tf2=.model_1.lleg*((7/.model_1.ts)**2)*(.model_1.newtheta+.model_1.thetas)+.model_1.lleg*(7/.model_1.ts)*.model_1.newdertheta-3813671.072*(.model_1.newtheta-.model_1.newgamma)
```

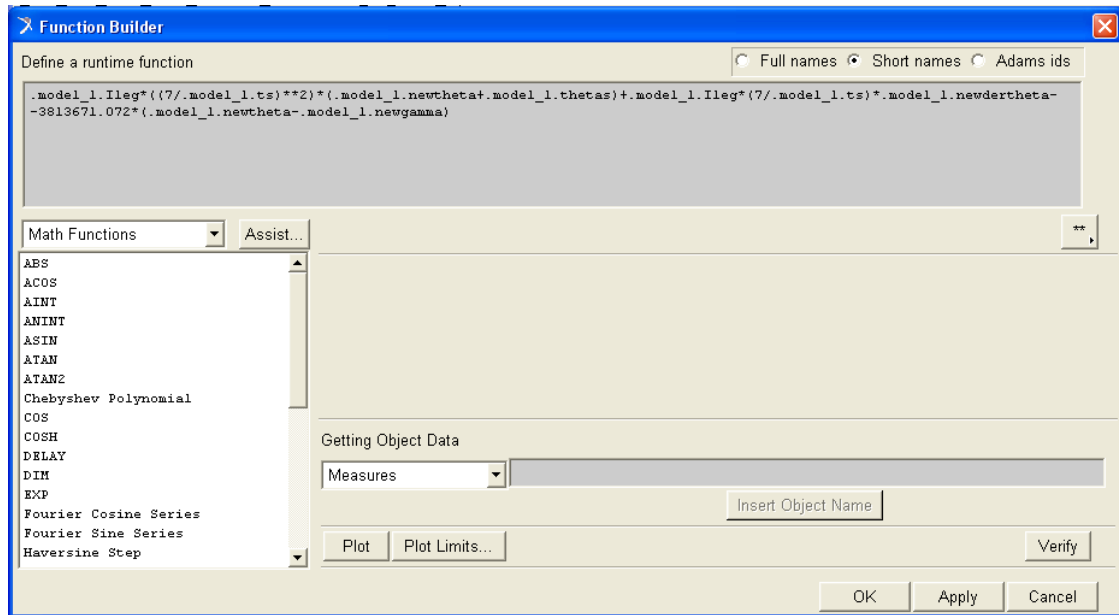


Figura 6.39 Creación de la función Tf2

Nueva pata apoyada

Th2=.model_1.newIrobot*((7/.model_1.ts)**2)*(-.model_1.newgamma-.model_1.thetas/2)-.model_1.newIrobot*(7/.model_1.ts)*.model_1.newdergamma-16000000*SIN(.model_1.newgamma)-3813687.32*SIN(.model_1.newtheta-.model_1.newgamma)-.model_1.Tf2_MEA_1

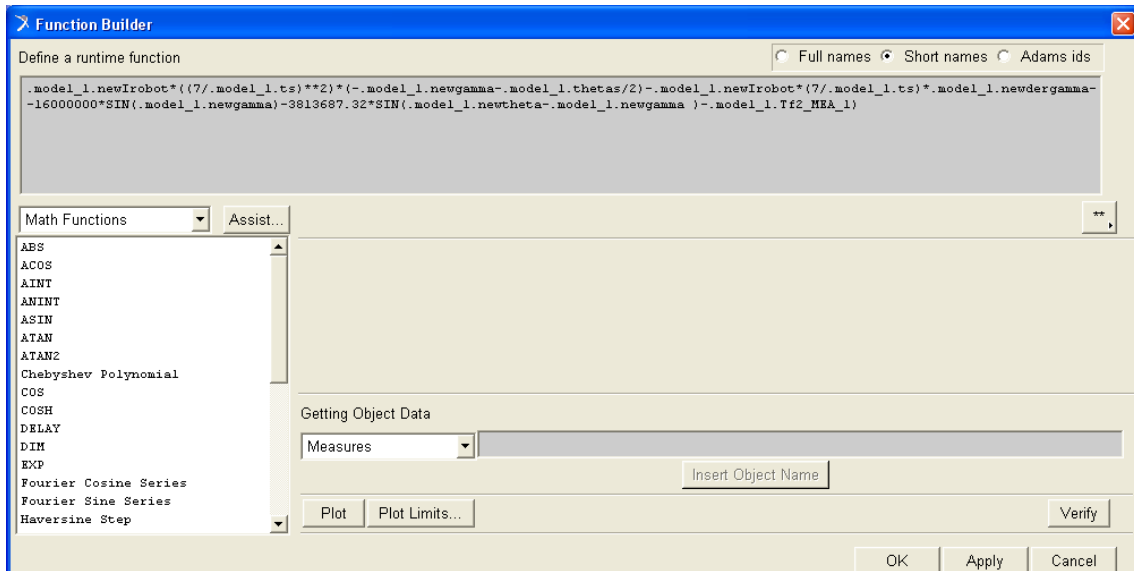


Figura 6.40 Creación de la función Th2

Y aplicados de la siguiente manera:

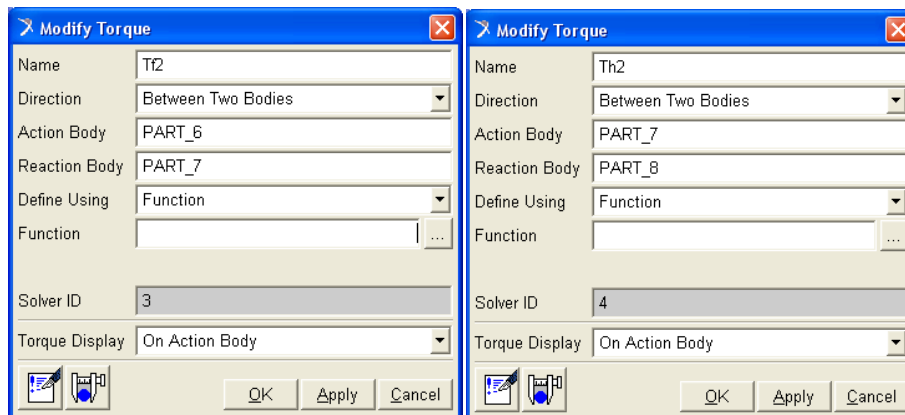


Figura 6.41 Especificaciones de los pares introducidos.

Una cuestión importante que presenta este método es definir el momento en el que se debe realizar el cambio de unos pares a otro.

Se pueden presentar varias soluciones en función de los objetivos que tiene nuestro movimiento, como sería realizar el cambio cuando se han alcanzado el periodo o la amplitud del paso. O establecer un sensor de contacto entre la pata y la plataforma de manera que cuando se produzca dicho contacto cambien los pares.

6.4.2 Doble paso con cambio en función del tiempo

Utilizar el tiempo como el factor determinante en el cambio de las ecuaciones de control no resulta muy preciso, ya que como se comprobó en el apartado de simulaciones con cambios de periodo, se producían errores elevados. Aunque de esta manera se ofrece la posibilidad de que el modelo sea el que se ajuste a dicho periodo, es decir, para el caso que estamos desarrollando de 1 segundo, si se le indica al modelo que cambie de ecuaciones con este tiempo, pero el periodo real es menor del indicado, el robot permanecerá quieto hasta que se haya cumplido el tiempo total. Se consigue de esta manera que el robot se ajuste al periodo teórico, aunque no lo haga estrictamente durante el movimiento.

La manera de introducir este tiempo en las ecuaciones de par es mediante el uso de condicionales. La instrucción If sigue la siguiente estructura:

IF(cond-A:B,C,D) donde:

A: condición para la cual está definida la función

B: función que debe realizar antes de la condición A

C: función que debe realizar durante la condición A

D: función que debe realizar después de la condición A

Por lo tanto se modifican las ecuaciones de par de la siguiente manera, donde B son las funciones de control explicadas en los apartados anteriores.

Tf y Th: If(time-1:B,0,0)

Tf2 y Th2: If(time-1:0,0,B).

Se desarrolla la simulación, prestando especial atención al momento de cambio en las ecuaciones.

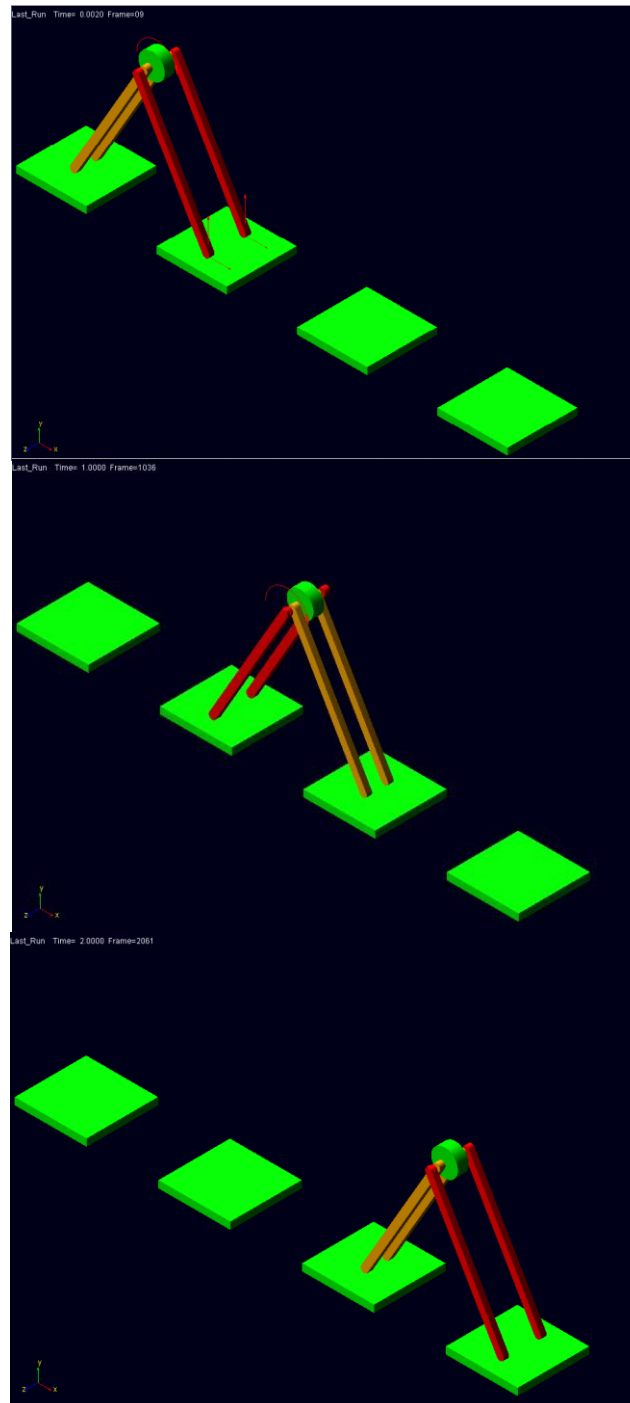


Figura 6.42 Simulación doble paso en función del tiempo.

El movimiento en la simulación se realiza de manera correcta, aunque es necesario destacar un importante aspecto. Se observa deslizamiento en dos fases diferentes del movimiento, por un lado entre la pata de apoyo y el suelo en el momento de comenzar cada paso, por la fuerza del par generado; por otro lado, cuando la pata en movimiento alcanza su plataforma de apoyo y se mantiene a la espera del cambio en las ecuaciones de control, se produce un pequeño deslizamiento hacia delante, incrementándose por lo tanto la amplitud con la que habían finalizado las patas su movimiento.

Una manera de comprobar cómo influyen estos pequeños deslizamientos en el correcto desarrollo de la simulación es observar la evolución temporal de los ángulos (figura 6.43).

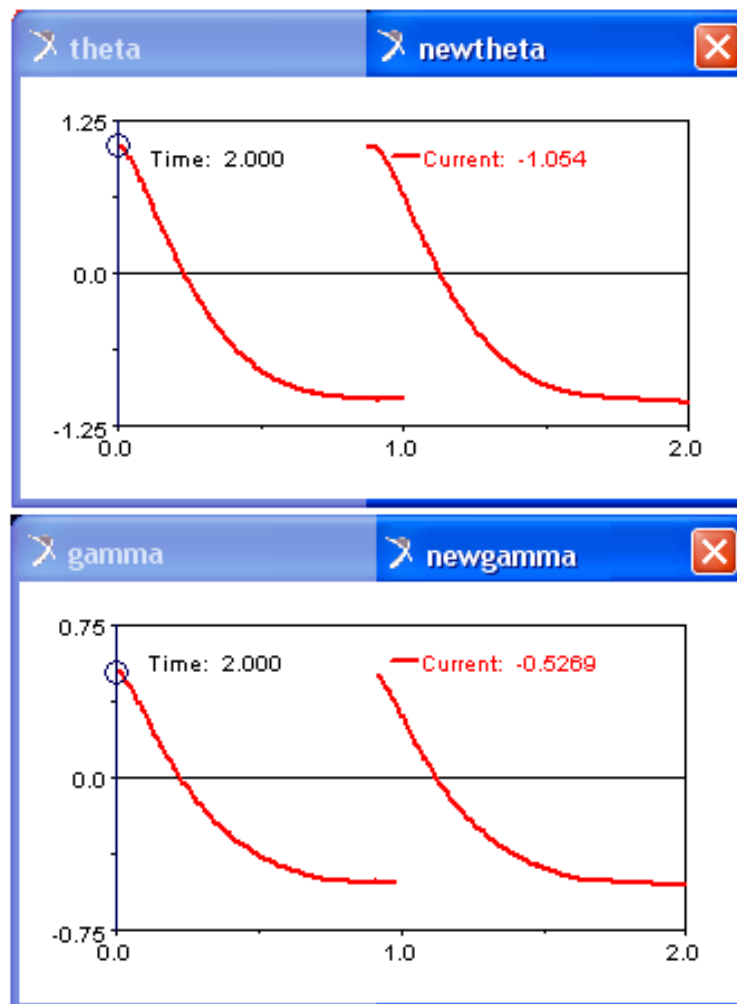


Figura 6.43 Evolución de ángulos en la simulación de dos pasos

Se comprueba que la evolución temporal que siguen los ángulos es igual tanto antes como después de realizarse el cambio en las ecuaciones de control cuando se alcanza un tiempo en la simulación de 1 segundo. Y analizando los valores alcanzados tenemos:

	Valor de control (rad)	Valor obtenido en la simulación (rad)	Error cometido (%)
Theta	1'0472	1'054	0'65
Gamma	0'5236	0'5269	0'63

Tabla 6.5 Cálculo de errores en simulación con dos pasos.

Los errores obtenidos son despreciables por lo que se puede asegurar que estos deslizamientos no influyen de manera relevante en el paso.

Otra medida con la que comprobar el correcto desarrollo del movimiento es conocer cuanta distancia recorre el robot. Para ello se suman las distancias recorridas en los dos pasos para un periodo total de 2 segundos, y se compara con la distancia teórica que debería recorrer.

La distancia total recorrida por el robot en dos pasos, según la medida realizada en el programa es de 1209'0 mm, mientras que el valor de control esperado eran 1200 mm.

Calculando el error cometido, este corresponde a un 0'75%, por lo que los deslizamientos observados son casi despreciables en comparación con el movimiento total, aun siendo los periodos ajustados diferentes a los reales.

CAPÍTULO 7

Conclusiones

7 CONCLUSIONES

En primer lugar se ha realizado el estudio y la comprensión de los algoritmos de control propuestos en el artículo [1], llegándose a detectar un error en la expresión esencial para el desarrollo de este proyecto. La capacidad más atractiva en este proyecto es la de construir pares o fuerzas como funciones de variables “actuales” (en el instante de cálculo), lo que permite simular tareas de control.

Se ha adquirido un manejo fluido de las características básicas, incluso de partes avanzadas, de la herramienta Adams View partiendo de un desconocimiento total de esta. Se trata de una herramienta de diseño, simulación y cálculo potente, que permite desarrollar infinidad de pruebas, aunque en algunos aspectos se muestra poco intuitiva para el usuario que la maneja.

Después de probar diversos tamaños de robot de una manera simplificada, se escogió el que resultaba mejor para las simulaciones que se llevarían a cabo. Una vez modelado el conjunto se introdujeron las ecuaciones de control, teniendo especial cuidado con los signos y la correcta definición de las funciones creadas. De esta manera se cumplía el objetivo inicial de generar un modelo que permitiese realizar varias simulaciones modificando diferentes parámetros en las ecuaciones de control.

Una vez creado el modelo se estudió su comportamiento en diferentes situaciones y movimientos. Simulaciones que arrojaron ciertos resultados de los que extraen estas conclusiones:

- Mala adaptación a los cambios de periodo: el modelo se ajusta mal a los cambios de periodo, se produce un sobrefrenado en el avance, que provoca que cuanto mayor es su valor menos avanza el robot, con la consecuencia de no completar el paso en su totalidad. En el modelo simulado se consiguen errores aceptables de menos del 15% con periodos menores de 1'25 segundos, por lo que es recomendable realizar simulaciones por debajo de ese valor.
- Buena adaptación a los cambios en la amplitud del movimiento: al contrario que sucedía con el periodo, la otra propiedad modificable por el usuario si funciona de manera correcta en el modelo. Para todos los ángulos introducidos de la amplitud de control, el movimiento se completa con apenas desviaciones del objetivo final.
- Necesidad de coeficientes de rozamiento altos: los pares elevados que se producen al comienzo del movimiento, generan grandes esfuerzos en los puntos de contacto entre las patas y el suelo. Por ello es conveniente trabajar con valores altos que eviten en la manera de lo posible los deslizamientos entre piezas, teniendo en consideración que a medida que se aumenta el periodo, el esfuerzo inicial generado es menor y por lo tanto también la necesidad de superficies con altos coeficientes de rozamiento.



Conocidas mediante los ensayos cuales eran las condiciones más favorables para realizar pasos con nuestro modelo, el siguiente objetivo desarrollado fue encadenar dos pasos de manera consecutiva, simulando así un movimiento de caminar.

Quedan así cubiertos los principales objetivos iniciales propuestos: comprensión y análisis del modelo y modelado y ensayo del robot mediante Adams View.



CAPÍTULO 8

Futuros trabajos

8 FUTUROS TRABAJOS

Se pueden realizar diversos trabajos posteriores que vendrían a continuar con las tareas desarrolladas en el presente proyecto.

La que parece más necesaria sería corregir el modelo físico o las ecuaciones de control de manera que los movimientos se adapten igual de bien a los cambios de periodos que a los cambios de amplitud. Un modelo que no limitase las simulaciones en ningún rango de trabajo constituiría una potente herramienta de ensayo.

Lo que constituiría un siguiente paso en la configuración del robot, sería definir con exactitud los elementos necesarios para su construcción real, es decir, modelos de actuadores, controladores, piezas...y demás elementos necesarios para un ensayo real.

Realizando estos ensayos reales se comprobaría si verdaderamente la herramienta de diseño utilizada arroja resultados equiparables a la realidad.

Un aspecto que resulta importante en estos robots es el consumo, la realización de un estudio energético, así como la implantación de medidas para mejorarlo sería de un gran interés.

En cuanto a las simulaciones de más de un paso, un aspecto mejorable es el establecimiento de un criterio de contacto adecuado para el intercambio de algoritmos entre las patas al final de cada paso.

CAPÍTULO 9

Bibliografía e infografía

9 BIBLIOGRAFÍA E INFOGRAFÍA

[1]: Mauricio Alba, Juan Carlos García Prada, Jesús Meneses, Higinio Rubio “Center of percussion and gait design of biped robots”

[2]: http://buscon.rae.es/draeI/SrvltConsulta?TIPO_BUS=3&LEMA=robot (15/7/2011)

[3]: <http://es.wikipedia.org/wiki/Robot> (15/7/2011)

[4]: Juan Álvaro Gallego, Arturo Forner Cordero, Juan C. Moreno, Edyta A. Turowska, Jose L. Pons Grupo de Bioingeniería - CSIC, “SÍNTESIS DE LA MARCHA ROBÓTICA MEDIANTE LA APROXIMACIÓN DEL CICLO LÍMITE: EL ROBOT BÍPEDO ESBiRRo” <http://www.cea-ifac.es/actividades/jornadas/XXIX/pdf/240.pdf> (15/7/2011)

[5]: http://www.humanoid.waseda.ac.jp/booklet/kato_2.html (16/7/2011)

[6]: http://e-archivo.uc3m.es/bitstream/10016/7716/5/PFC_Javier_Perez_Caballero.pdf (16/7/2011)

[7]: <http://webdiis.unizar.es/~neira/docs/CBalaguer-CEDI2007.pdf> (18/7/2011)

[8]: <http://www.takanishi.mech.waseda.ac.jp/top/research/wabian/index.htm> (19/7/2011)

[9]: Proyecto fin de carrera: “Estudio de la implantación de habilidades a un robot bípedo” desarrollado por Jorge Gallardo Mielgo. Universidad Carlos III de Madrid. Año 2007.



ANEXO

ARTICLE IN PRESS

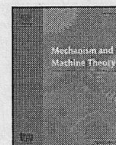
MAMT-01704; No of Pages 13

Mechanism and Machine Theory xxx (2010) xxx–xxx



Contents lists available at ScienceDirect

Mechanism and Machine Theory

journal homepage: www.elsevier.com/locate/mechmt

Center of percussion and gait design of biped robots

Mauricio Alba*, Juan Carlos Garcia Prada, Jesus Meneses, Higinio Rubio

Department of Mechanical Engineering, Universidad Carlos III de Madrid, Avenida de la Universidad 30, 28911 Leganes, Spain

ARTICLE INFO

Article history:
Received 9 November 2009
received in revised form 10 June 2010
accepted 12 June 2010
Available online xxxx

Keywords:
Biped robots
Linear control
Passive dynamics
Center of percussion
Inverted pendulum

ABSTRACT

In this paper a novel gait generation technique is proposed, whose objective is to optimize the mechanical design of actuated bipeds in order to reduce energy consumption. The novelty is the use of the *center of percussion* of the robot to calculate the equivalent simple pendulum of the system. With the equivalent pendulum, the gait is parameterized using its natural frequency. The advantage of this technique is the possibility to assess the influence of the mass and inertia properties of a biped robot on the gait. Therefore, an intelligent design of the kinematic chain can be made using the equivalence developed here. In addition, the method is a new option to linearize biped systems. Control is achieved by designing an adaptive PD controller for gait generation. This algorithm is applied to a compass-like biped with knees and a flywheel as a replacement for the body.

© 2010 Elsevier Ltd. All rights reserved.

1. Introduction

The control and design of mechanical systems are tasks which are closely related. The mechanical properties of the mechanism determine the possibilities for a control system to succeed in achieving its objective. Although controllers for biped robots consider their dynamics, either directly or indirectly, they offer little insight about the best solution for the mechanical design. Several articles illustrate the increasing interest of these systems from the mechanical point of view; [6] develops an interesting correlation between dynamics and energy expense.

Zero Moment Point (ZMP) based bipeds are among the most outstanding ones; the Honda Asimo, HRP-2, HOAP among others belong to this kind of robot. They can carry out complex tasks, and are widely used in research path planning. Furthermore, they are highly sophisticated and are often equipped with a large number of exteroceptive sensors. Representative examples can be found in [4,8]. The major drawback of these robots is their energy consumption, they can work continuously for 40 min to 1 h when fully charged.

Tad McGeer demonstrated that it is possible to generate highly anthropomorphic gait with no actuation [11]. Passive dynamic machines were not able to walk on level ground, but only on a slope. The modification of McGeer machines with some actuation were developed to walk on level ground. Some authors use heuristic methods as described in [7,20]. They calculate the energy loss in each step in order to recover it by means of a state machine and different types of actuators, such as pneumatic or electric ones. Although there is no true control, there is sensing to make the walking cycle more robust. The major advantage of these robots is their low energy consumption.

Other authors have developed more sophisticated techniques to imitate McGeer machines. With control algorithms, they simulate the conditions of a fully passive walker as described in [1–3,21]. Their objective is gait synthesis instead of robust walking.

The work previously described is not able to link the algorithms used with the mechanical parameters of the robot in order to optimize gait. The systems shown in [9,10,16] use the classic linear inverted pendulum model (LIMP) or improvements thereof,

* Corresponding author. Tel.: +34 91 624 8380; fax: +34 91 624 9430.

E-mail addresses: dalba@ing.uc3m.es (M. Alba), jcgprada@ing.uc3m.es (J.C.G. Prada), meneses@ing.uc3m.es (J. Meneses), hrubio@ing.uc3m.es (H. Rubio).

0094-114X/\$ – see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.mechmachtheory.2010.06.008

Please cite this article as: M. Alba, et al., Center of percussion and gait design of biped robots, Mech. Mach. Theory (2010), doi:10.1016/j.mechmachtheory.2010.06.008

ARTICLE IN PRESS

2

M. Alba et al. / Mechanism and Machine Theory xxx (2010) xxx–xxx

such as the angular momentum pendulum model (AMPM). They use the mass parameters of the robot actively to synthesize gait. In this way, they produce a balance between the low energy consumption of passive machine based systems, and the robustness present in high energy spender ZMP robots. The overall idea in these walkers is to orbit around a limit cycle, and reject perturbations by forcing the cycle to go into the right values within the space state.

Pendulum-like mathematical models provide an intuitive description of the complexity of a biped robot. Despite their wide popularity, their use is limited to some parts of the walking cycle. In this article, a new approach to find an equivalence between a biped system and an inverted pendulum is defined. The equivalent pendulum is characterized by an equivalent point mass, located at the center of percussion (CPe) of the system. The pendulum is assumed to have actuation at the pivot point, located at the ZMP.

With this definition, the gravitational and dynamic forces can be compensated, and thus the damped natural frequency of the actuated pendulum is used to generate the gait. Considering this, future robots can be designed taking into account the effects of their mass distribution on the natural frequency of the equivalent pendulum. This allows optimization of the best cruise speed for a particular design.

Among the most impressive dynamic walkers are the ones designed by Raibert in [17]. The objective of this project is to follow a similar path, starting from simple planar models and then scaling them into more sophisticated 3D designs. For that reason, the simplest planar biped is used in this article, making several simplifications such as a flywheel instead of a body and point feet to simplify gait.

2. Equivalent dynamic systems

A system is considered dynamically equivalent to another when it can be described by the same equations of motion. For a rigid body this problem is widely studied, and a common approach is to replace the rigid body by two point masses. In order to allow dynamic equivalence three equalities have to be satisfied between the systems:

- The total mass
- The center of mass
- The moment of inertia

These three conditions lead to a system of 3 equations with 3 unknowns. The complete development of the equations can be found in any book of theory of machines and mechanisms as for example [19].

This method allows replacing the rigid body of Fig. 1 by two point masses m_A and m_B , located at distances l_A and l_B respectively. For l_A given the value of l_B , m_A and m_B is calculated as follows:

$$l_B = \frac{I_G}{m l_A} \quad (1)$$

$$m_A = m \frac{l_B}{l_A + l_B} \quad (2)$$

$$m_B = m \frac{l_A}{l_A + l_B} \quad (3)$$

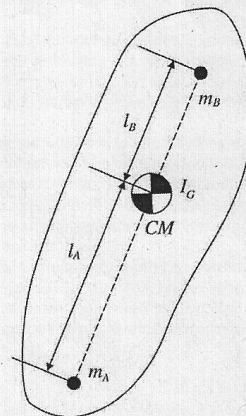


Fig. 1. A rigid body can be replaced by two point masses m_A and m_B , both systems have the same dynamics.

Please cite this article as: M. Alba, et al., Center of percussion and gait design of biped robots, Mech. Mach. Theory (2010), doi:10.1016/j.mechmachtheory.2010.06.008

ARTICLE IN PRESS

M. Alba et al. / Mechanism and Machine Theory xxx (2010) xxx–xxx

3

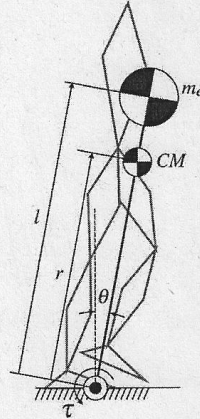


Fig. 2. Biped with its equivalent inverted pendulum; l is the distance from the ZMP to the CPe, r is the distance from the pivot to the CM. Notice that the position of the pivot is at the ZMP, also called the center of pressure. The other equivalent mass at the ZMP is not shown because the ZMP is assumed fixed.

The location of m_B is known as the center of percussion of the system. If a rigid body is restrained to rotate around m_A , the last can be eliminated from the dynamics of the system (although it should be used for the statics) and only m_B is considered. In this work the location of m_A is on the ZMP, and m_B is used to construct an equivalent pendulum.

3. Equivalent inverted pendulum of a biped robot

The following analysis has been developed assuming the floor is flat and even. This is required because under these conditions ZMP has the same location of the center of pressure [18]. In this context, center of pressure refers to the distance-weighted average location of the individual pressures on the foot. More details are described in [15]. However, in simple terms, the center of pressure is the point where the distributed force on the sole of the foot can be replaced by a point force of the same magnitude.

Fig. 2 shows a biped with its equivalent simple pendulum. The pendulum has length l and a point mass m_e , which is aligned with the line formed by the ZMP and the center of mass (CM). m_e is calculated to maintain the equivalence of mass and inertial properties between the pendulum and the whole biped, just as shown in Eq. (3). The distance l corresponds to the location of m_e , and also to the (CPe) as explained in Section 2.

This point is pivot dependent; it varies according to the position of the selected center of oscillation. Its use in walking robots results from the fact that the CPe is the length of the equivalent simple pendulum, the one having the same period as the original system. The CPe can be calculated using the moment of inertia I_{ZMP} around the ZMP, the mass m_{robot} of the system, and the distance r from the pivot to the center of mass; repeating Eq. (1) with robot parameters:

$$l = \frac{I_{ZMP}}{rm_{robot}} \quad (4)$$

Notice that in Fig. 2 the mass m_A calculated in Section 2 has not been drawn. This is possible if the system rotates around the location of m_A . Therefore, in order to make Eq. (4) valid, the phZMP should be in a fixed position during the stance phase.

Imagining the biped of Fig. 2 hung upside down from its ZMP, the period of oscillation of this system is the same as the simple pendulum with length l , according to the definition of CPe. Then the equation of motion of the biped at this exact instant can be expressed as:

$$m_e l^2 \ddot{\theta} + m_e g \sin \theta = 0 \quad (5)$$

Eq. (5) does not describe the biped in Fig. 2. In order to do so, two terms are included, the torque τ due to the actuation of the motors, and the torque τ_d due to the dynamic effects of the moving links. The resulting equation is:

$$m_e l^2 \ddot{\theta} + m_e g \sin \theta + \tau + \tau_d = 0 \quad (6)$$

τ is the input control of the system. Although it seems difficult to apply torque to this point, the fact is that any torque applied to the robot is reflected in its ZMP. To illustrate this fact, Fig. 3 shows the dynamic equivalence between an inverted pendulum with

Please cite this article as: M. Alba, et al., Center of percussion and gait design of biped robots, Mech. Mach. Theory (2010), doi:10.1016/j.mechmachtheory.2010.06.008

ARTICLE IN PRESS

4

M. Alba et al. / Mechanism and Machine Theory xxx (2010) xxx–xxx

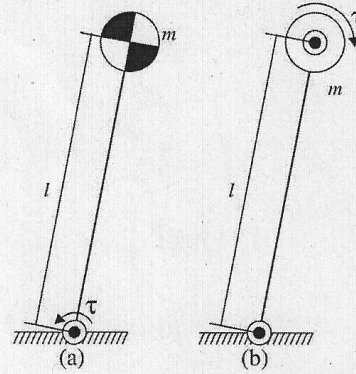


Fig. 3. (a) Inverted pendulum (b) Inverted pendulum with a flywheel. Both models have the same dynamics.

torque applied to its pivot, and an inverted pendulum with a free pivot but with a flywheel instead. The details of this deduction can be found in [15].

Taking this into account the term τ can be written as the summation of the torques of each articulation. Considering Fig. 4, it is written:

$$\tau = -\sum \tau_i \quad (7)$$

Eq. (7) relates all the torques with the effect they produce on the ZMP. This equation describes the effects of the motors of the robot over its equilibrium, thus adding intelligent restriction to make better gait design.

4. Description of the model

The equations of the previous section can be applied to any biped, but at this point several simplifications will be made. The model can be seen in Fig. 5, where the body has been replaced by a flywheel. It then becomes a source of torque by changing its angular momentum in such a way that the body orientation is not a variable to control. Each leg is connected independently but concentrically to the flywheel, providing the model with a bisecting hip. Knees have been added to allow correct foot clearance,

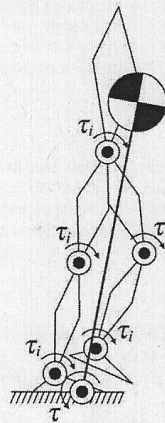


Fig. 4. Torques τ_i at the articulations of the biped. The sum is equivalent to torque τ of the equivalent pendulum.

Please cite this article as: M. Alba, et al., Center of percussion and gait design of biped robots, Mech. Mach. Theory (2010), doi:10.1016/j.mechmachtheory.2010.06.008

ARTICLE IN PRESS

M. Alba et al. / Mechanism and Machine Theory xxx (2010) xxx–xxx

5

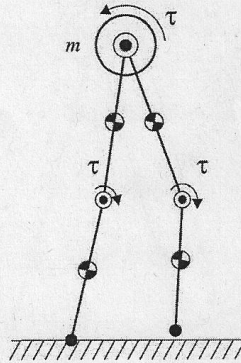


Fig. 5. Flywheel biped described in this paper. Each motor of the legs can apply torque to the flywheel independently. The knees have been added to perform foot clearance.

providing two degrees of freedom to each leg (hip and knee), so the whole model has a total of four degrees of freedom of movement.

The feet are modeled as one contact point, allowing an exact location of the ZMP when the robot is being supported by one foot. The location of the ZMP on a foot with distributed contact (for example, a planar one) is easily computed, but for the sake of simplicity and for transparency of the algorithms, the one point solution was chosen. Besides point feet fix the position of the ZMP as required in Section 3.

Ankle torques contribute to the equilibrium of a biped [5], but ankles are fragile elements. Changing the equilibrium strategy will contribute to relieve the stress in this articulation. Point feet cannot exert torque directly onto the floor, but instead they rely heavily on the dynamic equivalence shown in Fig. 3. This is coherent with the fact that the human body has its most powerful muscles in the upper part of the legs. In fact, dance and creative movement instructors teach their pupils how to balance with their hips and not with their ankles.

Fig. 6 shows variables and control inputs. γ is the angle between the shin and the normal vector to the floor. θ is the angle between legs and is measured from the stance leg, which means that its sign changes after each step. α is the angle of the knee and is different from zero only when the leg is swinging for foot clearance.

τ_f , τ_h and τ_k are the control inputs. τ_f is the torque necessary to actuate the swinging leg, while τ_h is the input control for body attitude and is produced by the stance leg over the flywheel. Just as with the angles, their definition depends on which leg is the stance one at that time. Finally τ_k is the knee actuation.

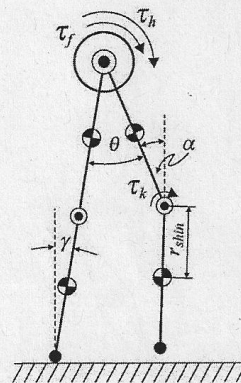


Fig. 6. Angles of the flywheel biped: θ angle between legs, α angle of the knee joint, γ angle between the stance leg and the normal vector of the floor. τ_h is the torque between the stance leg and the flywheel, and τ_f then one between the swinging leg and the flywheel.

Please cite this article as: M. Alba, et al., Center of percussion and gait design of biped robots, Mech. Mach. Theory (2010), doi:10.1016/j.mechmachtheory.2010.06.008

ARTICLE IN PRESS

6

M. Alba et al. / Mechanism and Machine Theory xxx (2010) xxx-xxx

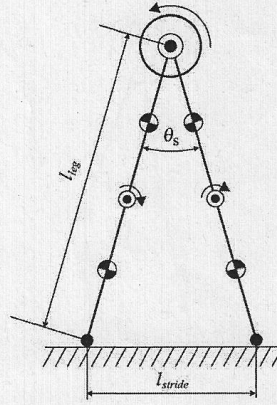


Fig. 7. Some parameters of the gait: l_{leg} is the length of the leg, θ_s is the angle of the legs at the double stance phase, and l_{stride} is the length of the stride. The robot is in the double stance position.

5. Gait generation

Gait can be divided into two stages. The first one is when only one leg is in contact with the floor, and the other is when the robot is in double stance phase. However, several restrictions will be applied to produce gait in order to simplify the double stance phase. The first restriction is that the stance leg has to be in a straight position all the time. Consequently, the swinging leg will be straight before touching the floor, and then become the new stance leg. That will put the robot in the position shown in Fig. 7, making the double stance phase take place within an infinitesimal space of time. This situation is similar to the synthetic wheel described in [11]. Another advantage of this approach is that only one algorithm is required, the one for the single stance phase.

This algorithm will be divided into two tasks: one to control body attitude and another for the swinging leg. The idea is to compensate the major sources of nonlinearities in order to apply linear controllers for both cases. A state machine switches the controller when the swinging leg becomes the new stance leg.

5.1. Gait parameters

Walking is not a regular task. In fact, complex limit cycles are described in the study of passive dynamics. However, despite its complexity, the periodicity quality makes it susceptible to parameterizing, thereby simplifying the description of the gait.

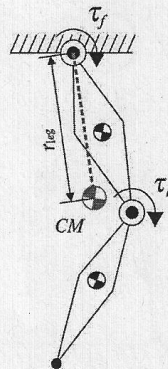


Fig. 8. Swinging leg. τ_f is the torque at the hip, τ_k is the torque at the knee, and r_{leg} is the distance from the hip to the CM.

Please cite this article as: M. Alba, et al., Center of percussion and gait design of biped robots, Mech. Mach. Theory (2010), doi:10.1016/j.mechmachtheory.2010.06.008

ARTICLE IN PRESS

M. Alba et al. / Mechanism and Machine Theory xxx (2010) xxx–xxx

7

Fig. 7 shows the robot during double stance phase. This position is repeated at the beginning of every cycle. Thus, the angle θ_s defines aperture between the legs before the double stance phase. This can define the stride length with simple trigonometry. The period t_s of each step is also of interest. In combination with θ_s the gait is completely described for regular terrain. Because the double stance phase is assumed to be infinitesimal, then t_s accounts for the total time of a single step. With this in mind, more intuitive quantities can be estimated such as the average walking velocity.

$$v_{avg} = \frac{l_{stride}}{t_s} \quad (8)$$

5.2. Swinging leg control

In order to generate gait correctly, the robot is supported by one leg, while the other leg reaches its position to become the new stance leg. During this time the swinging leg performs two tasks: the first is to position the thigh in front of the body, i.e. to reach angle θ_s shown in Fig. 7; the second is to bend the knee to avoid foot scuffing. Swinging leg parameters are shown in Fig. 8.

The swinging leg has to reach its position in a time less than or equal to the period t_s of the gait. To ensure that the leg is ready to support the robot a shorter time is chosen. A fixed fraction of the period is defined for the time required to swing the leg:

$$t_f = ct_s \quad (9)$$

where $0 \leq c \leq 1$ is a constant selected to leave enough safety time gap. If the leg gets hung at a fixed point it will behave like a pendulum. Because it is actuated, it is possible to design a PD controller once the appropriate objectives have been selected. The control law for this controller is:

$$\tau_f = k_p(\theta - \theta_s) + k_d \dot{\theta} - m_{leg} |\vec{r}_{leg} \times \vec{g}| \quad (10)$$

The first and second terms of Eq. (10) are the proportional and derivative parts of the controller. The third term is a gravity compensation term to get rid of nonlinearities introduced by the large angle between the legs. To avoid foot scuffing the knee needs to be bent just at a small angle, and therefore the dynamic effects of knee movements are neglected.

k_p and k_d are selected to produce critical damped movement on the leg. Thus the solution for the equation of movement is $\theta(t) = (A + Bt)e^{-\omega_0 t}$; ω_0 is the natural frequency of the system when k_d is equal to zero. The inverse of ω_0 is the time constant τ of the system, an important quantity to measure the stabilization of linear systems. A common criterion is to use 7 times τ as the stabilization time [13], thus this is the time required to swing the leg:

$$t_f = 7\tau \quad (11)$$

$$t_f = \frac{7}{\omega_0} \quad (12)$$

ω_0 is defined as:

$$\omega_0 = \sqrt{\frac{k_p}{I_{leg}}} \quad (13)$$

Where I_{leg} is the moment of inertia of the leg around the hip. Eqs. (12) and (13) are combined to find k_p :

$$k_p = I_{leg} \left(\frac{ct_s}{7} \right)^2 \quad (14)$$

Now k_d can be defined in terms of k_p and the damping ratio ζ . The last one is defined as:

$$\zeta = \frac{k_d}{2\sqrt{k_p I_{leg}}} \quad (15)$$

As was mentioned previously in this section, the system is designed to have critical damping, which is interpreted as ζ having a value of 1. From Eq. (15), k_d is calculated:

$$k_d = 2\sqrt{k_p I_{leg}} \quad (16)$$

Please cite this article as: M. Alba, et al., Center of percussion and gait design of biped robots, Mech. Mach. Theory (2010), doi:10.1016/j.mechmachtheory.2010.06.008

ARTICLE IN PRESS

8

M. Alba et al. / Mechanism and Machine Theory xxx (2010) xxx–xxx

Because knees were added to prevent foot scuffing, their control is carried out by coupling the movements with angle γ . The equation can then be written:

$$\tau_k = k_{kp}(\alpha - f(\gamma)) + k_{kd}\dot{\alpha} - m_{\text{shin}}|\vec{r}_{\text{shin}} \times \vec{g}|$$

where k_{kp} is a stiffness value, and k_{kd} is a viscous friction term. $f(\cdot)$ bends the knee most of the time, and then blocks it near the end of the swinging of the leg. Many functions satisfy these conditions. The function chosen was Ae^{-bx} where A and b are constants conveniently selected for the system. Better $f(\cdot)$ can be designed considering the whole kinematics of the robot. A good criterion could be constant foot clearance in a region of interest, however this is beyond the scope of this paper, and the selected function worked fine in the simulations.

Eq. (10) is valid if the knee is blocked; because the movement of the shin is small, their dynamic effects have not been considered. However, the actuation of the knee is an easy term to compensate. Therefore, Eq. (10) is modified as follows:

$$\tau_f = k_p(\theta - \theta_s) + k_d\dot{\theta} - m_{\text{leg}}|\vec{r}_{\text{leg}} \times \vec{g}| - \tau_k \quad (17)$$

5.3. Stance leg control

The swinging leg was controlled using an adaptive PD controller. The values of the proportional and derivative gain are recalculated in real time according to the changes in mass and inertia properties. These changes are produced by the reconfiguration of the leg when swinging, because of knee bending. The actuation to control the swinging leg is applied directly to the hip, which is the center of the arc, to measure the controlled variable θ .

The stance leg is the one responsible for body attitude, i.e. the angle γ between the stance leg and the floor, as described by Fig. 6. However, direct actuation on the support point is not possible. Even for robots with extended feet this is a major problem because the torque required to control the body only with the ankle joint, is too large. That is the reason for the dynamic equivalence mentioned in Section 3 and illustrated in Fig. 3.

The stance leg only applies torque over the flywheel and this torque is reflected on the support point, also defined as the ZMP, as an opposite reaction. The objective is to reach the double stance phase; as shown in Fig. 7, in this position γ has a value of $\theta_s/2$.

Controlling body attitude is quite similar to controlling the equivalent inverted pendulum of the system, i.e. using Eq. (6). But some changes are necessary. First, the variable to measure is not the angle of the equivalent pendulum, but angle γ , and only reactions τ_f and τ_k , due to the movement of the swinging leg, are considered. Thus, the torque at the hip joint applied by the stance leg is:

$$\tau_h = k_{po}(\gamma - \theta_s/2) + k_{do}\dot{\gamma} - m_{\text{robot}}|\vec{r}_{\text{robot}} \times \vec{g}| - \tau_f - \tau_k \quad (18)$$

The equation of motion of the system according to Eq. (6) is rewritten to match the parameters of the robot. $m_d l^2$ is the inertia of the robot I_{robot} with respect to the ZMP, $m_d g l \sin \theta$ is the torque of the gravitational force applied to the CoM, τ is the summation of joint torques τ_h , τ_f and τ_k , τ_d are the torques produced by centrifugal and Coriolis accelerations, and θ is replaced by γ . The terms k_{po} and k_{do} are the gains of the controller. After applying all the changes the expression is as follows:

$$I_{\text{robot}}\ddot{\gamma} + m_{\text{robot}}|\vec{r}_{\text{robot}} \times \vec{g}| + \tau_h + \tau_f + \tau_k + \tau_d = 0 \quad (19)$$

In order to find the equations of movement of the system, Eq. (18) is replaced in Eq. (19). The gravitational term is cancelled, and only τ_d remains. The final equation is:

$$I_{\text{robot}}\ddot{\gamma} + m_{\text{robot}}|\vec{r}_{\text{robot}} \times \vec{g}| + k_{po}(\gamma - \theta_s/2) + k_{do}\dot{\gamma} - m_{\text{robot}}|\vec{r}_{\text{robot}} \times \vec{g}| + \tau_d = 0 \quad (20)$$

Analyzing this last expression we can conclude that except for τ_d , this equation describes a damped harmonic oscillator. If the term τ_d is small enough to be neglected, a similar method to the one used for the swinging leg controller is used to calculate the gains k_{po} and k_{do} . Analogous to Eq. (14), but with the difference that the criteria is taken based on the whole period of the walking cycle t_s instead of the fraction ct_s , k_{po} is expressed as:

$$k_{po} = I_{\text{robot}} \left(\frac{t_s}{7} \right)^2 \quad (21)$$

Please cite this article as: M. Alba, et al., Center of percussion and gait design of biped robots, Mech. Mach. Theory (2010), doi:10.1016/j.mechmachtheory.2010.06.008

ARTICLE IN PRESS

M. Alba et al. / Mechanism and Machine Theory xxx (2010) xxx–xxx

9

In the same way k_d is calculated, and its expression is:

$$k_{do} = 2\sqrt{k_{po}I_{robot}} \quad (22)$$

5.4. Putting it all together

As described previously, the walking algorithm requires only one controller for the single stance phase because the double stance happens in an infinitesimal time period. Control was divided into two tasks: one for body attitude and another to swing the leg. The swinging leg was set up to perform its task, in a shorter time than that required for the stance leg to reach its position. The final position of the stance leg, just before becoming the swinging one, is such that the robot has to end up as shown in Fig. 6, i.e. the angle γ reaches $\theta_s/2$.

In order to walk, once the swinging leg is in front of the robot, it becomes the new stance leg, and the former stance leg starts to swing. To do so, the controllers have to switch in order to perform the required task according to the situation. Then, a state machine is used. When the swinging leg touches down on the floor, the controllers are switched, and the cycle repeats itself to produce gait. There is significant amount of work on these systems and details about such machines can be found in [7] and [17].

6. Simulation results

The algorithms were tested with Modelica[®] using Dymola[®] implementation. Their elements are object oriented. Hence, highly detailed models can be made with minor assumptions. The implementation of such models is done in a bond graph fashion, and new models or extensions thereof can be made using Modelica[®] language.

The models are made mostly with the rigid body dynamics library developed in [14]. This library is not designed to simulate walking robots, but new elements were created to do so. In addition, the actuated joints have limited usability for the purposes of this article, and thus, modifications of these elements were developed. The details of the programming of the model are not provided here, but they are not necessary for interpretation of the data.

The chosen mass of the different parts was 1 kg, and the length of the shin and thigh were 0.35 m and 0.37 m respectively. The mass of the shin and thigh are assumed as a point mass located at the center of each member. The flywheel has an inertia of 0.01 kg m². Table 1 shows this information.

The multibody dynamics library cannot handle contact. In order to simulate linkages or similar situations, an anchor point should be defined first. This is not the case for walking robots. They rely on the interaction between feet and floor because the contact point changes in every step. To solve this problem a new element was created. This element simulates point contact using a damped mass-spring model. It works by determining the coordinates x_{td} and z_{td} for the contact point when the y coordinate goes below 0. Then tangential and normal forces are calculated. This can be written as:

$$f_{ty} = \begin{cases} \text{if } y_{ft} \leq 0 \text{ then } k_t y_{ft} + b_t \dot{y}_{ft} \\ \text{else } 0 \end{cases} \quad (23)$$

For the tangential reaction force:

$$f_{tx} = \begin{cases} \text{if } y_{ft} \leq 0 \text{ then } k_t (x_{ft} - x_{td}) + b_t \dot{x}_{ft} \\ \text{else } 0 \end{cases} \quad (24)$$

In other words, what this model of contact does is determine the plane $y = 0$ as the surface of the floor, and then it calculates the reactions with an elastic model with damping. z_{td} is not described here because the simulation is restricted to the plane. The knee lock is simulated in a similar way. A reactive torque is triggered when the angle of the knee goes below a certain value:

$$\tau_{lock} = \begin{cases} \text{if } \alpha \leq 0 \text{ then } k_{kr} \alpha + b_{kr} \dot{\alpha} \\ \text{else } 0 \end{cases} \quad (25)$$

This approach of elastic contact solves the problem of chattering, commonly found when impact-based contact models are implemented. Furthermore, it provides a softer and more realistic behavior.

Table 1
Parameters of the robot used in the simulation.

Mass of each part	1.00 kg
Inertia of the flywheel	0.01 kg m ²
Length of the shin	0.35 m
Length of the thigh	0.37 m
Total mass of the robot	5.00 kg

Please cite this article as: M. Alba, et al., Center of percussion and gait design of biped robots, Mech. Mach. Theory (2010), doi:10.1016/j.mechmachtheory.2010.06.008

ARTICLE IN PRESS

10

M. Alba et al. / Mechanism and Machine Theory xxx (2010) xxx-xxx

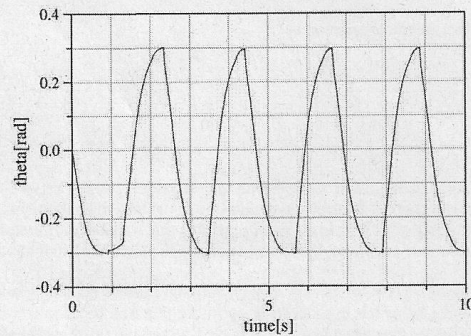


Fig. 9. θ follows the assigned value of 0.3 rad quite closely, although the period of 1.5 s is less precise.

The walking parameters were chosen as $\theta_s = 0.3$ rad and $t_f = 1.5$ s. The walking algorithm was designed to have critical damping, and works with a state machine. The state machine makes the controllers work as an on-off system. It is therefore expected to behave as these controllers. Fig. 9 shows the plot of angle θ , and shows its behavior, which is similar to the differential gap found in on-off systems.

Analyzing the walking parameters, it can be found that angle θ_s is followed with remarkable precision. The period t_s has a discrepancy showing an average of 1.09 s. This discrepancy is the consequence of nonmodeled phenomena, principally the dynamic effects due to the swinging leg. Variation of gait parameters showed similar results, i.e. it closely followed θ_s with an appreciable variation of t_s . Carrying out a deeper analysis, the swinging leg controller which is less prone to dynamic perturbations, is responsible for the value of θ_s . However, t_s is regulated by the stance leg controller, and withstands the dynamic effects of the whole robot.

Knee impacts due to locking of the knees cause no visible deviation of the expected performance, as can be observed in Fig. 10. The controllers act as variable springs as can be seen in Eqs. (7) and (10); then they can naturally reject perturbations. Other researchers have experimented with springs to obtain a more robust gait as for example [7,15]. They have softened and improved the original McGeer models by adding springs and elastic actuation, but the adaptive nature of the controllers developed in this paper provides a better disturbance rejection than those developed in previous works.

The plot of angle α also shows a smooth behavior. Fig. 11 shows this plot. The flat part at zero degrees is when the leg is swinging. Once it makes contact, it goes up to approximately 0.15 rad, which is the value of $\theta_s/2$, and the lift-off occurs at about the same angle, but with a negative value. This plot shows how precisely the stance leg controller reaches the commanded angle $\theta_s/2$, as described by Eq. (18). The perturbation due to non modeled dynamics only affects the period, but not the precision of the set point of the controller.

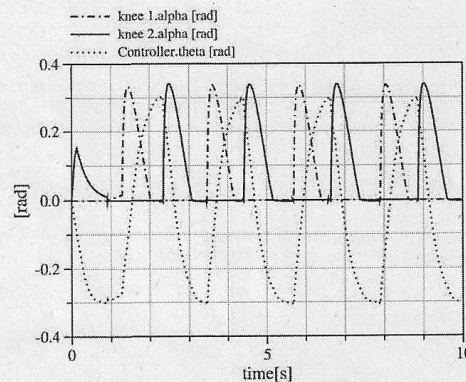


Fig. 10. Plot of the angles of the knees plus θ , after the first step lock impacts cause no effects on the system.

Please cite this article as: M. Alba, et al., Center of percussion and gait design of biped robots, Mech. Mach. Theory (2010), doi:10.1016/j.mechmachtheory.2010.06.008